DATA SCIENCE

Neighborhoods Analysis for Restaurant - Manchester

2021

FEBRUARY 21

IBM CAPSTORNE PROJECT -

Authored by: DCJ

Business issue

Intro

Manchester is a city and metropolitan borough in Greater Manchester, England. The city has the country's fifth-largest population at 547,627 (as of 2018) and lies within the United Kingdom's second-most populous urban area, with a population of 2.7 million, third most-populous county, at around 2.8 million, and second-most populous metropolitan area, with a population of 3.3 million. It is fringed by the Cheshire Plain to the south, the Pennines to the north and east, and an arc of towns with which it forms a continuous conurbation. The local authority for the city is Manchester City Council.

The city is notable for its architecture, culture, musical exports, media links, scientific and engineering output, social impact, sports clubs and transport connections. Manchester Liverpool Road railway station was the world's first inter-city passenger railway station. At the University of Manchester, Ernest Rutherford first split the atom in 1917, Frederic C. Williams, Tom Kilburn and Geoff Tootill developed the world's first stored-program computer in 1948, and Andre Geim and Konstantin Novoselov isolated the first graphene in 2004.

"The purpose of this study is to offer a potential investor valuable information about the city of Manchester"

Always have seen Manchester as a very interesting city to run a business and in particular a restaurant given the data that have shared above. Also, I have a special interest in the area given that I am a very big football fan and some of my favorite teams are from this particular area. The purpose of this study is to offer a potential investor valuable information about the city of Manchester.

So for this my main objective is to have a look at the data that we have available in the web and what I have learned through this course to be able to perform an analysis given the right and adequate data about the main neighborhoods that exist nowadays in Manchester using the techniques that there exist about machine learning and data analysis and hopefully this information would be of use to a potential investor in the area.

The data that I used

Neighborhoods data

As I learned during the course, some valuable sources could be used for this purpose using web scraping with BeautifulSoup library in Python. The extracted info is found in the following link: 'https://en.wikipedia.org/wiki/Category:Areas_of_Manchester'

Geographical coordinates

Whith GeoPy I extracted the geographical coordinates of several neighborhoods of the city I was interested in. This is important of course to have a coordinate system well defined for your project and also to have georeferenced the areas you would like to start your analysis. The following was the starting data:

```
In [14]: man_data.head()
```

Out[14]:

	Neighbourhood	Latitude	Longitude
0	Baguley	53.399090	-2.285610
1	Barlow Moor	53.422201	-2.246093
2	Belle Vue, Manchester	42.955853	-71.459019
3	Benchill	53.381730	-2.261250
4	Beswick, Manchester	53.478390	-2.200320

Venue data

As well, as I learned during the course, I had to extract the venue data and the best way to do this was through the use of the API of FourSquare. This venues were used to analyze the data of the areas or neighborhoods I was interested in and helped me to come to some conclusions that later on this document I will expose.

The methodology that I used: Feature extraction

One-hot encoding maps a column of label indices to a column of binary vectors, with at most a single one-value. This encoding allows algorithms which expect continuous features, such as Logistic Regression, to use categorical features. As I learned during the course, I determined that through this method I could finally have a venue for each row and that each column will contain the frequency of occurrence of that particular category. In the sense that if I had each feature as a category that belongs to a venue that then would be converted into binary. And then all venues would be grouped as neighborhoods, as follows:

In [21]:
```
man_1hot = pd.get_dummies(explore_man[['Venue Category']], prefix="", prefix_sep="")


# Add neighbourhood column back to dataframe
man_1hot['Neighbourhood'] = explore_man['Neighbourhood']


# Move neighbourhood column to the first column
fixed_columns = [man_1hot.columns[-1]] + man_1hot.columns[:-1].values.tolist()
man_1hot = man_1hot[fixed_columns]

man_1hot.head()
```

Out[21]:

Unsupervised learning

As you may know, k-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. k-means clustering minimizes within-cluster variances (squared Euclidean distances), but not regular Euclidean distances, which would be the more difficult Weber problem: the mean optimizes squared errors, whereas only the geometric median minimizes Euclidean distances. For instance, better Euclidean solutions can be found using k-medians and k-medoids.

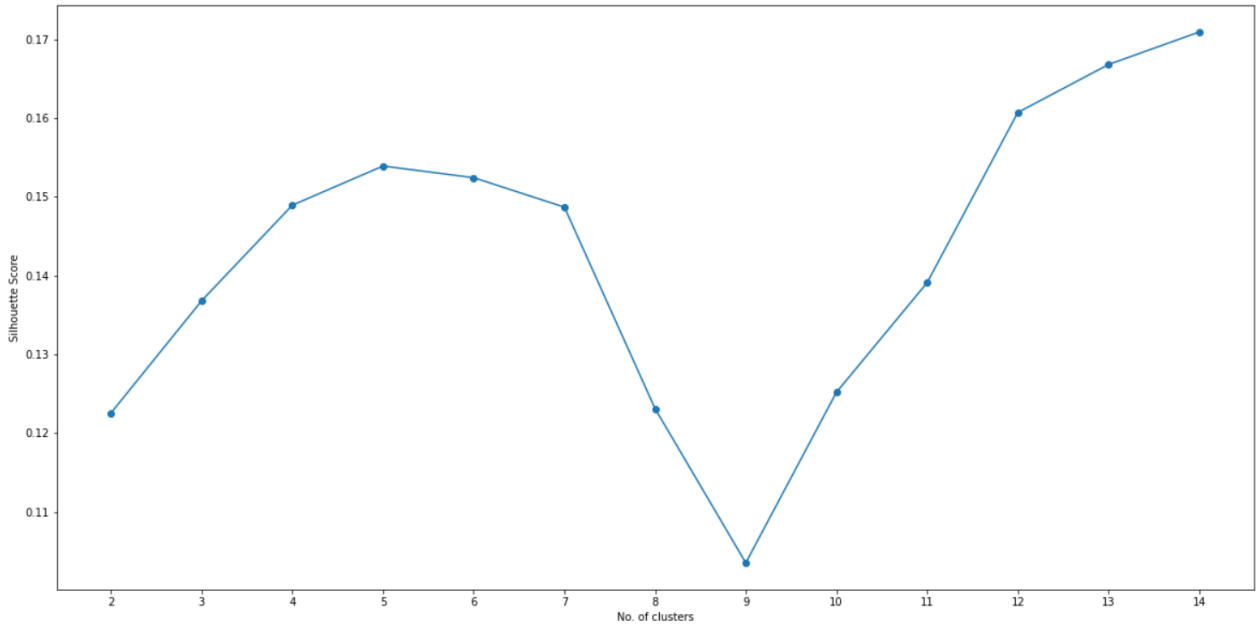
The problem is computationally difficult (NP-hard); however, efficient heuristic algorithms converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both k-means and Gaussian mixture modeling. They both use cluster centers to model the data; however, k-means clustering tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes.

The unsupervised k-means algorithm has a loose relationship to the k-nearest neighbor classifier, a popular supervised machine learning technique for classification that is often confused with k-means due to the name. Applying the 1-nearest neighbor classifier to the cluster centers obtained by k-means classifies new data into the existing clusters. This is known as nearest centroid classifier or Rocchio algorithm.

In this sense, I used K means for this purpose and to use unsupervised learning to find out similarities between neighborhoods as follows:

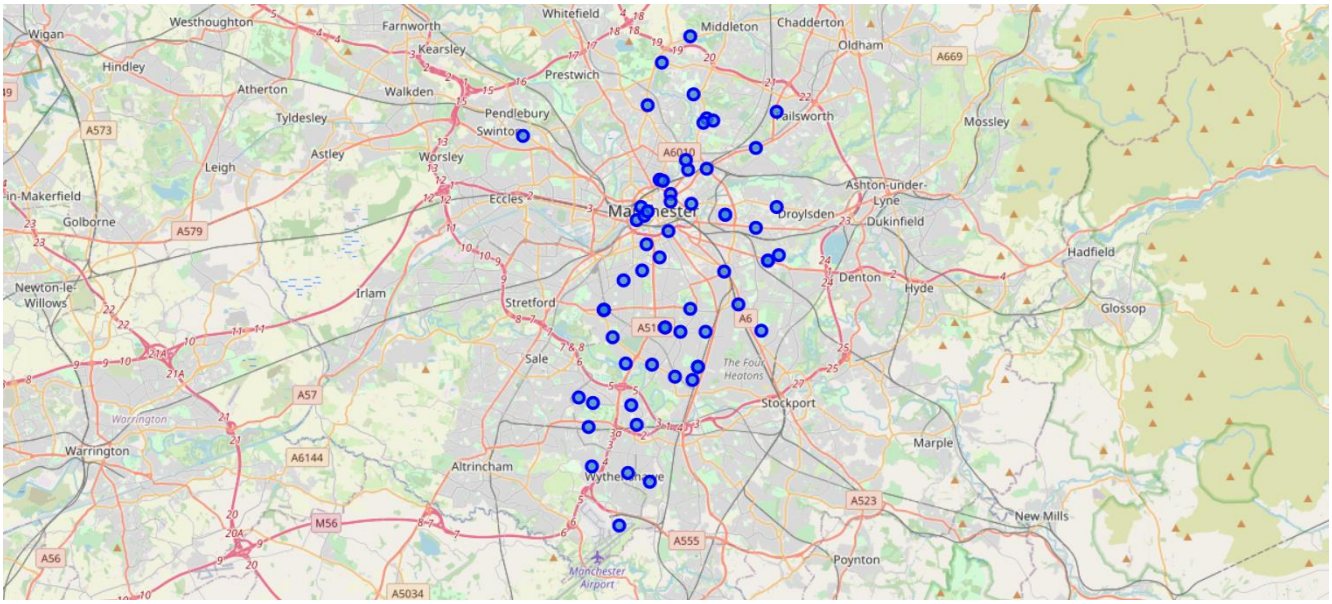
In [27]:
```
max_range = 15 #Max range 15 (number of clusters)

from sklearn.metrics import silhouette_samples, silhouette_score


indices = []
scores = []

for man_clusters in range(2, max_range) :

    # Run k-means clustering
    man_gc = man_grouped_clustering
    kmeans = KMeans(n_clusters = man_clusters, init = 'k-means++', random_state = 0).fit_predict(man_gc)

    # Gets the score for the clustering operation performed
    score = silhouette_score(man_gc, kmeans)

    # Appending the index and score to the respective lists
    indices.append(man_clusters)
    scores.append(score)
```

In [28]:
```
plot(max_range, scores, "No. of clusters", "Silhouette Score")
```



Plotting

Next the plotting was carried away in order to visualize data to give me an understanding of how to be able to compare data and start drawing some conclusions. Folium was used to plot the maps.

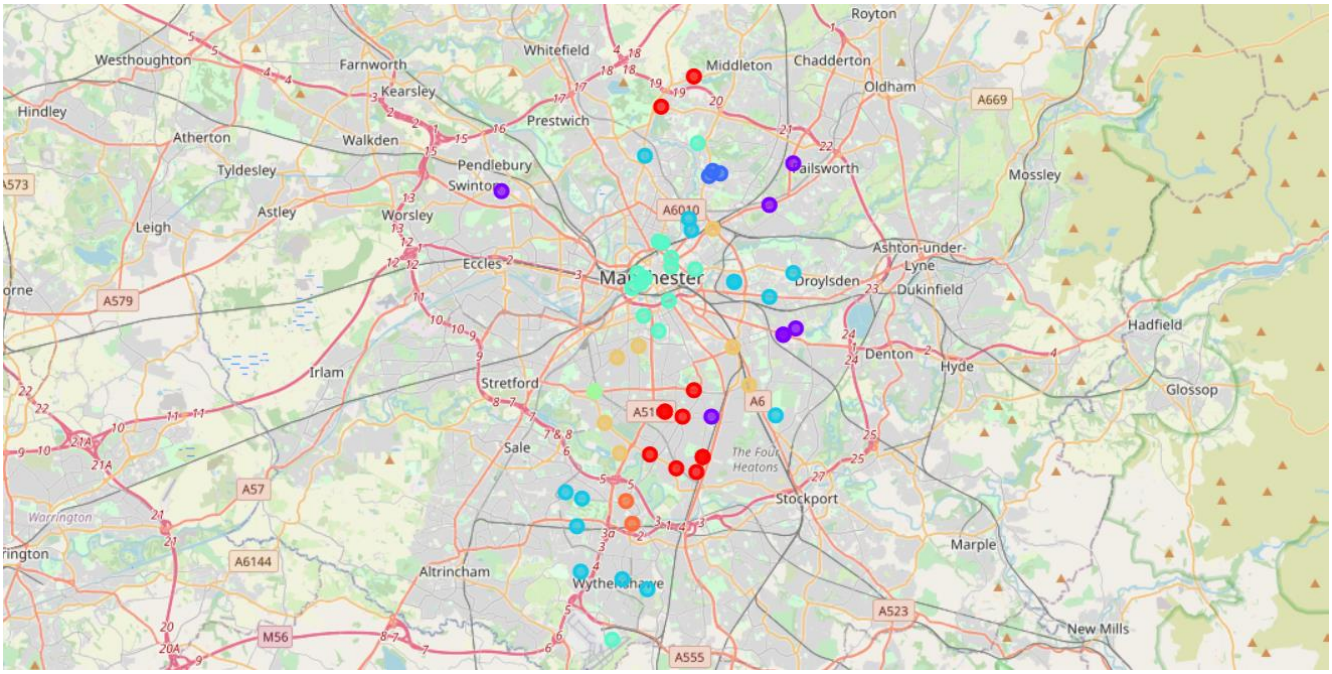


Results

As mentioned above, all the steps were followed to be able to get to this point in which the clusters was identified and as for the plotting of the clusters I applied the following: .

In [33]:
```
map_clusters = folium.Map(location=[latitude, longitude], zoom_start=11)

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

markers_colors = []
for lat, lon, poi, cluster in zip(man_final['Latitude'], man_final['Longitude'], man_final['Neighbourhood'],
                                  man_final['Cluster Labels']):
    label = folium.Popup(str(poi) + ' (Cluster ' + str(cluster + 1) + ')', parse_html=True)
    map_clusters.add_child(
        folium.features.CircleMarker(
        [lat, lon],
        radius=5,
        popup=label,
        color=rainbow[cluster-1],
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7))

map_clusters
```

With the following visualization of the clusters:



Business discussion and conclusions

- The client that would like to carefully decide and have into account all other factors to take a decision. Data analysis and machine learning were the principles to assist on decision making.
- After studying all of the four clusters it is recommendable to have an interest in setting up a restaurant and the best possibilities would be Barlow Moore or Brooklands, that fall into cluster four and if somebody is interested in opening a restaurant this information would be useful. For this conclusions, Python's inbuilt libraries such as BeautifulSoup, Folium and GeoPy were key to be able to determine.
- As mentioned, cluster 4 would be very attractive for opening a restaurant.
- The K-Means model worked really well and was successful in the clustering exercise.