# IBM Data Science Project

# INDEX

# Introduction

Manchester is one of the biggest cities in the UK. It lies within the United Kingdom's second-most populous urban area, with a population of 2.9 million and third-most populous metropolitan area, with a population of 3.3 million.

The population of Manchester comprises of people of various ethnicities from all over the world. The city is full of cafes, bars, lunchrooms, etc. Serving thousands of hungry customers, every day. The diversity in the population of the city has brought in a vast diversity in drinks and food habits of people.

In this project we will study the neighbourhoods and make recommendations accordingly.

## Business Problem

Our client is an investor who is interested in investing in a cafe in Manchester. They have approached us to study the market and suggest a location in one of the neighborhoods which would be in the best interest of the business. Our main objectives of this project would be to extract and analyze the right data about various neighborhoods of Manchester using various data science techniques and recommend our client a fitting location for their business.

# Data

In order to achieve our final goal we will need the following data:

· Neighborhoods of Manchester.
· Geographical coordinates of the neighborhoods.
· Venue data from FourSquare.

## Neighborhoods Data

This data was extracted from Areas of Manchester Wikipedia page (https://en.wikipedia.org/wiki/Category:Areas_of_Manchester) using web scraping with BeautifulSoup library in Python. This will give us a detailed list of neighborhoods present in Manchester.

## Geographical Coordinates

Later, the geographical coordinates of various neighborhoods were extracted using GeoPy library in Python. Geographical coordinates are necessary for plotting maps during the project for visualizing our data. After using GeoPy we added two columns to our dataframe with latitude and longitude information of each neighborhood as shown below:

| | Neighbourhood | Latitude | Longitude |
|---|---|---|---|
| 0 | Baguley | 53.399090 | -2.285610 |
| 1 | Barlow Moor | 53.422164 | -2.245970 |
| 2 | Belle Vue, Manchester | 42.955859 | -71.459019 |
| 3 | Benchill | 53.381730 | -2.261250 |
| 4 | Beswick, Manchester | 53.478390 | -2.200320 |

## Venue Data from FourSquare

Later we extracted venue data using FourSquare API. This venue data was used to study the venues in various neighborhoods in Manchester. This data provided important details of various food businesses in the area and helped us understand the competition. This data was very important because it helped us draw the main conclusion of the project.

# Methodology

## Feature Extraction

Feature extraction was carried out through One Hot Encoding. In this method, each feature is a category that belongs to a venue which is then converted into binary, this means that 1 means this category is found in the venue and 0 means the opposite. Then, all the venues are grouped by the neighborhoods, computing at the same time the mean. This will give us a venue for each row and each column will contain the frequency of occurrence of that particular category.

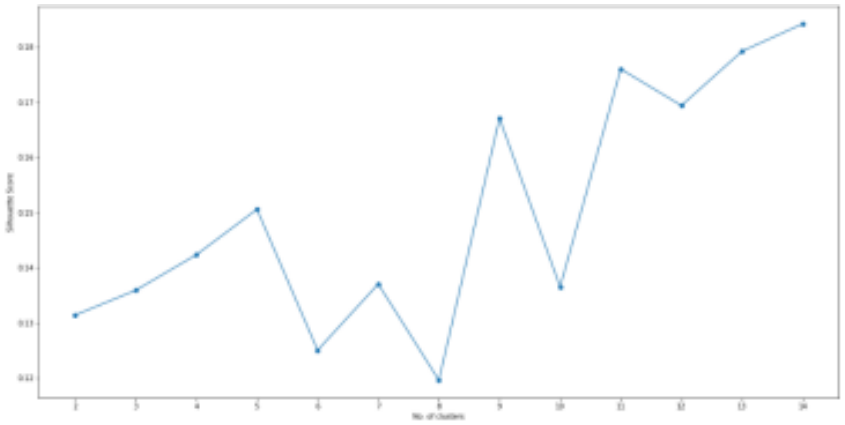
```
man_1hot = pd.get_dummies(explore_man[['Venue Category']], prefix="", prefix_sep="")


# 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()
```

## Unsupervised Learning

Unsupervised learning was carried out in order to find out the similarities between found similarities between neighborhoods. K-Means, a clustering algorithm, was implemented. In this case K-Means is used due to its simplicity and its similarity approach to find patterns.

- **K-Means:** K-Means is a clustering algorithm. This algorithm searches clusters within the data and the main objective function is to minimize the data dispersion for each cluster. Thus, each group found represents a set of data with a pattern inside the multi-dimensional features. It is necessary for this algorithm to have a prior idea about the number of clusters since it is considered an input of this algorithm. For this reason, the elbow method is implemented. A chart that compares error vs number of clusters is done and the elbow is selected. Then, further analysis of each cluster is done.

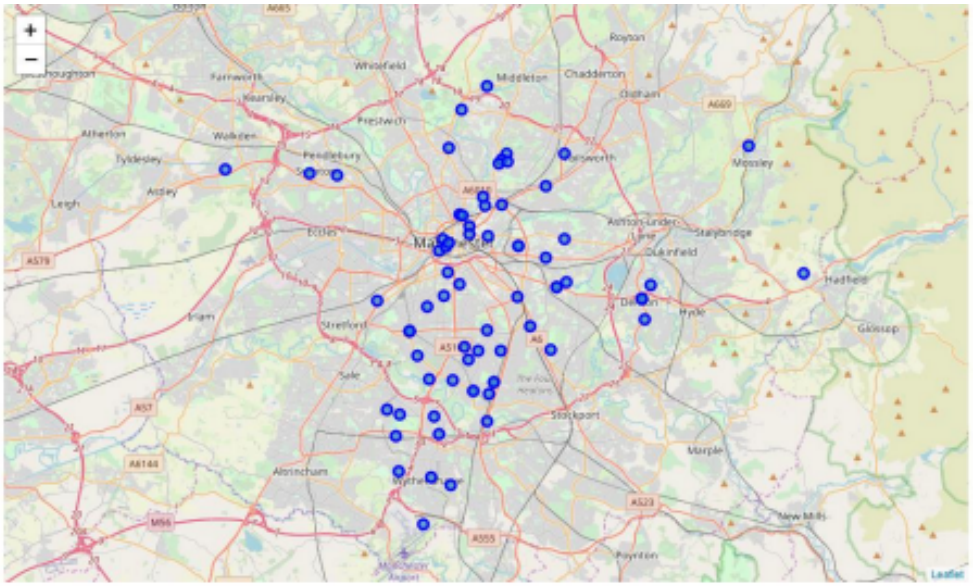
```
max_range = 15 #Max range 15 (number of clusters)

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)
```



## Plotting

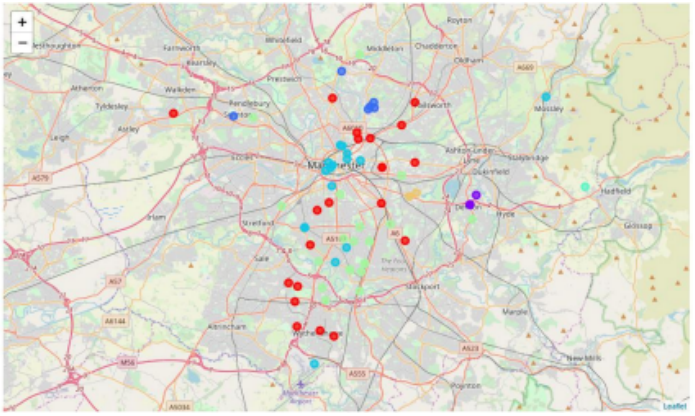
Various plotting techniques we used as well in order to visualize the data. Visualizing data often gives a clear understanding of the data as it is easier to spot patterns in a visualized data as compared to quantitative data.

- Folium: This library was used to plot maps of Manchester city as well as neighborhoods. Folium was also used to visualize the cluster data.



# Results

The above-mentioned, K-Means clustering method was applied to the dataframe of neighborhoods of Manchester city. As mentioned earlier the number of clusters that was derived from the elbow method was 8. The code as well as plotting of clusters can be seen below:



```
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
```

After visualizing the clusters, the individual clusters were studied and some important conclusions were derived. The neighborhood that had the most number of business was cluster number 4.

## Discussion

As mentioned earlier the most suitable neighborhoods for starting the business are present in cluster number 4. Our K-Means model worked perfectly and successfully  clustered similar neighborhoods together.

After studying all four clusters, it is recommended to the client that neighborhoods such as Barlow Moor, Brookelands and Hyde Newton (Ward) that fall in cluster 4 look like good locations for starting their cafe business. The client can go ahead and make a decision depending on other factors like availability and legal requirements that are out of scope of this project.

# Conclusion

Data analysis and machine learning techniques used in this project can be very helpful in determining solutions of certain business problems. Python's inbuilt libraries such as GeoPy, Folium and BeautifulSoup make it very easy and effective for a data scientist to analyze a geographical location because these libraries make it very easy to extract data that is easily available online. In this project we studied the neighborhoods of Manchester city and came up with a recommendation of neighborhoods where our client can start their new business.