The Battle of Neighborhoods –Coimbatore

Applied Data Science Capstone by IBM on Coursera

Jayasankar S

1. INTRODUCTION: BUSINESS PROBLEM

This project deals with the major venue categories in the neighborhoods of **Coimbatore, The Manchester of South India**. This project would specifically help business personal plan to start new Restaurants in Coimbatore, Tamil Nadu, India.

The **Foursquare API** is used to access the venues in the neighborhoods. Since, it returns less venues in the neighborhoods. Then they are clustered based on their venues using Data Science Techniques. Here the **k-means clustering algorithm** is used to achieve the task. The optimal number of clusters can be obtained using silhouette score metrics.

Folium visualization library can be used to visualize the clusters superimposed on the map of Coimbatore city. These clusters can be analyzed to help small scale business owners select a suitable location for their need such as Hotels, Shopping Malls, Restaurants or even specifically Indian restaurants or Coffee shops.

The major Target Audience would be small-scale business owners and stake holders planning to start their business at a location in Coimbatore. This project would help them find the optimal location to open a restaurant in the surroundings that has less number or no restaurants.

2. DATA REQUIREMENTS

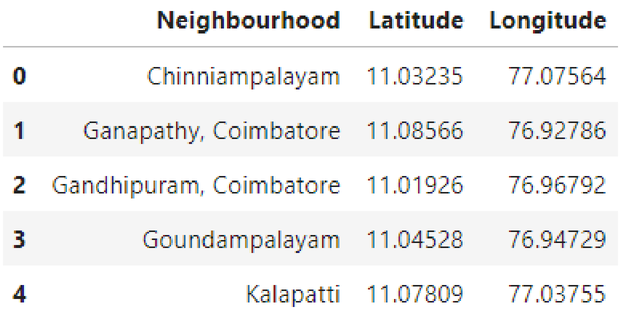
Neighbourhood data

Coimbatore has multiple neighbourhoods. **Wikipedia** has a dataset which has the list of Neighbourhoods in Coimbatore. There are a total of 36 neighbourhoods. The data are obtained from the Wikipedia page through **Data Scraping**

- https://en.wikipedia.org/wiki/Category:Neighbourhoods_in_Coimbatore

Geographical Coordinates

Later, the geographical coordinates of various neighbourhoods 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 neighbourhood as shown below:



	Neighbourhood	Latitude	Longitude
0	Chinniampalayam	11.03235	77.07564
1	Ganapathy, Coimbatore	11.08566	76.92786
2	Gandhipuram, Coimbatore	11.01926	76.96792
3	Goundampalayam	11.04528	76.94729
4	Kalapatti	11.07809	77.03755

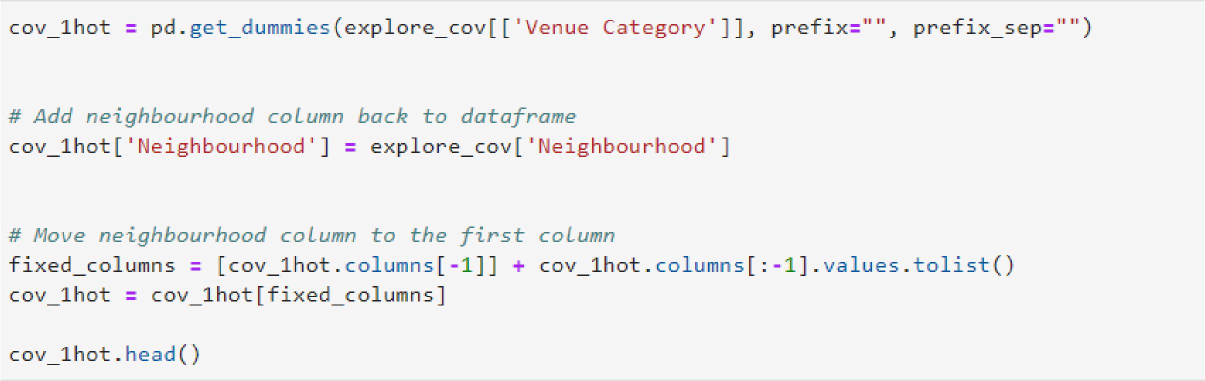
Venue Data from FourSquare

We extracted venue data using FourSquare API. This venue data was used to study the venues in various neighbourhoods in Coimbatore. This data provided important details of various restaurants 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.

3. 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 neighbourhoods, 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.

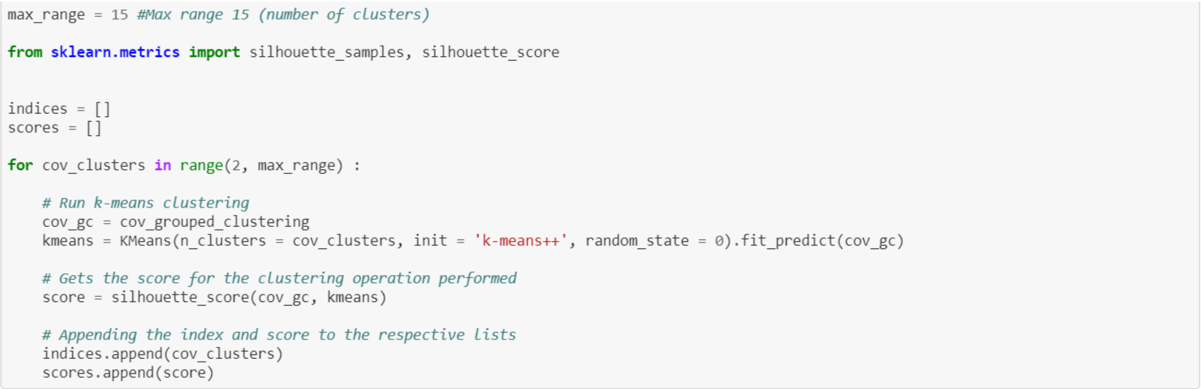


```
cov_1hot = pd.get_dummies(explore_cov[['Venue Category']], prefix="", prefix_sep="")


# Add neighbourhood column back to dataframe
cov_1hot['Neighbourhood'] = explore_cov['Neighbourhood']


# Move neighbourhood column to the first column
fixed_columns = [cov_1hot.columns[-1]] + cov_1hot.columns[:-1].values.tolist()
cov_1hot = cov_1hot[fixed_columns]

cov_1hot.head()
```

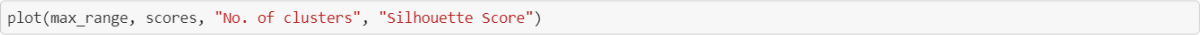
Unsupervised Learning

Unsupervised learning was carried out in order to find out the similarities between found similarities between neighbourhoods. **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.

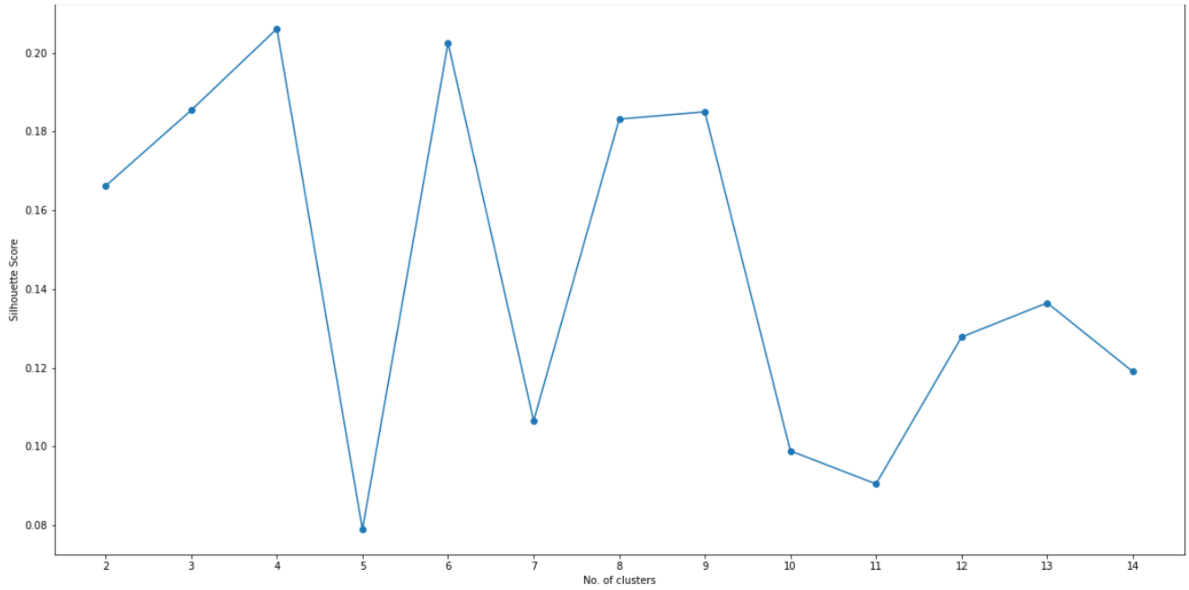
- **K-Means**: K-Means is a clustering algorithm. This algorithm search clusters within the data and the main objective function is to minimize the data dispersion for each cluster. Thus, each group found represents a set of data with a pattern inside the multi-dimensional features. It is necessary for 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 cluster 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 cov_clusters in range(2, max_range) :

    # Run k-means clustering
    cov_gc = cov_grouped_clustering
    kmeans = KMeans(n_clusters = cov_clusters, init = 'k-means++', random_state = 0).fit_predict(cov_gc)

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

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



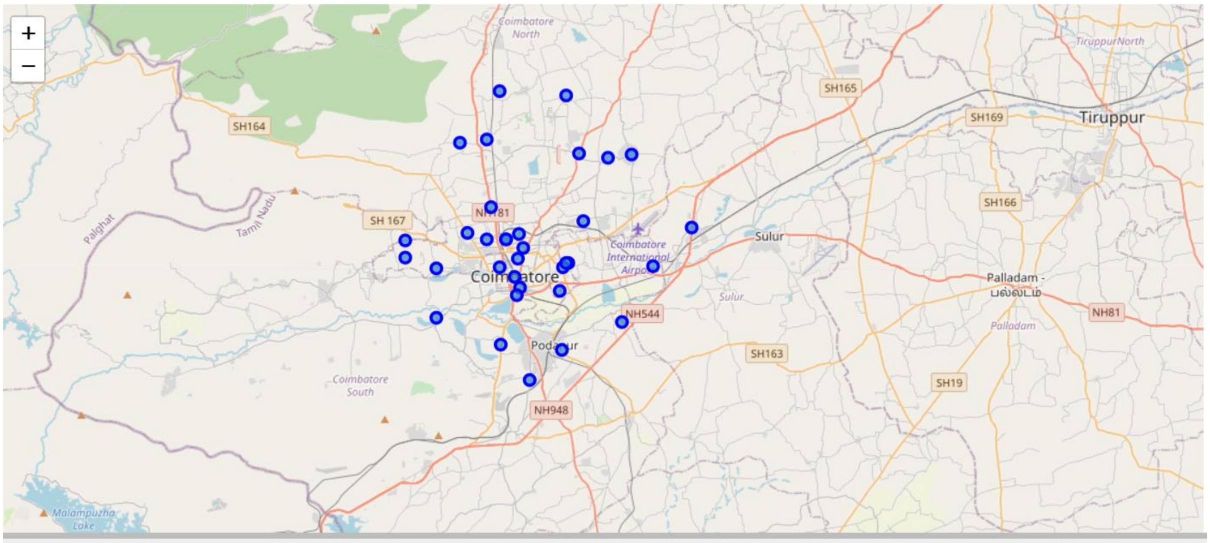
```
plot(max_range, scores, "No. of clusters", "Silhouette Score")
```



Plotting

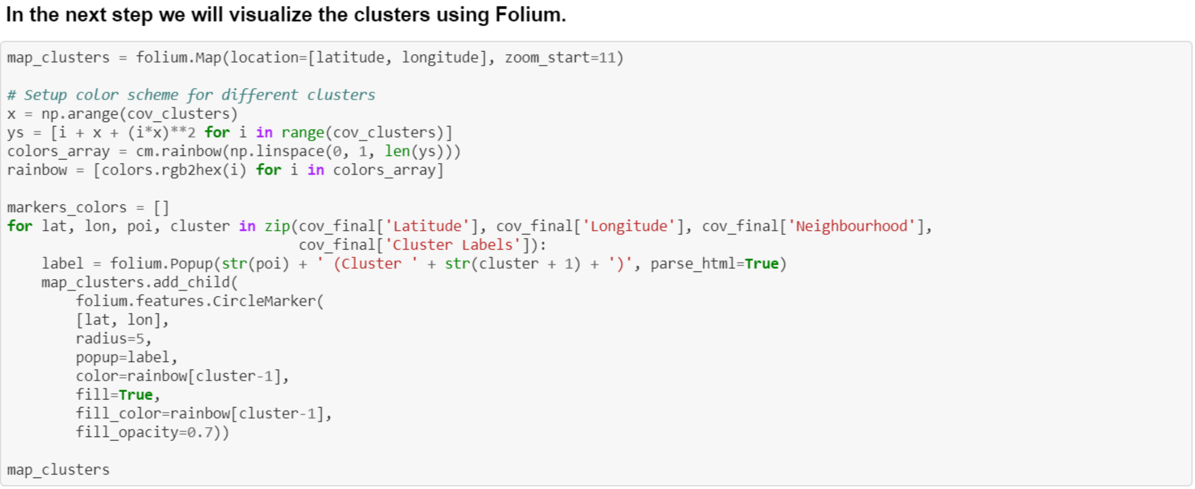
Various plotting techniques we used as well in order to visualize the data. Visualizing data often gives a clear understanding of the data as it is easier to spot patterns in a visualized data as compares to quantitative data.

- **Folium:** Folium library was used to plot maps of Coimbatore city as well as neighbourhoods. Folium was also used to visualize the cluster data.



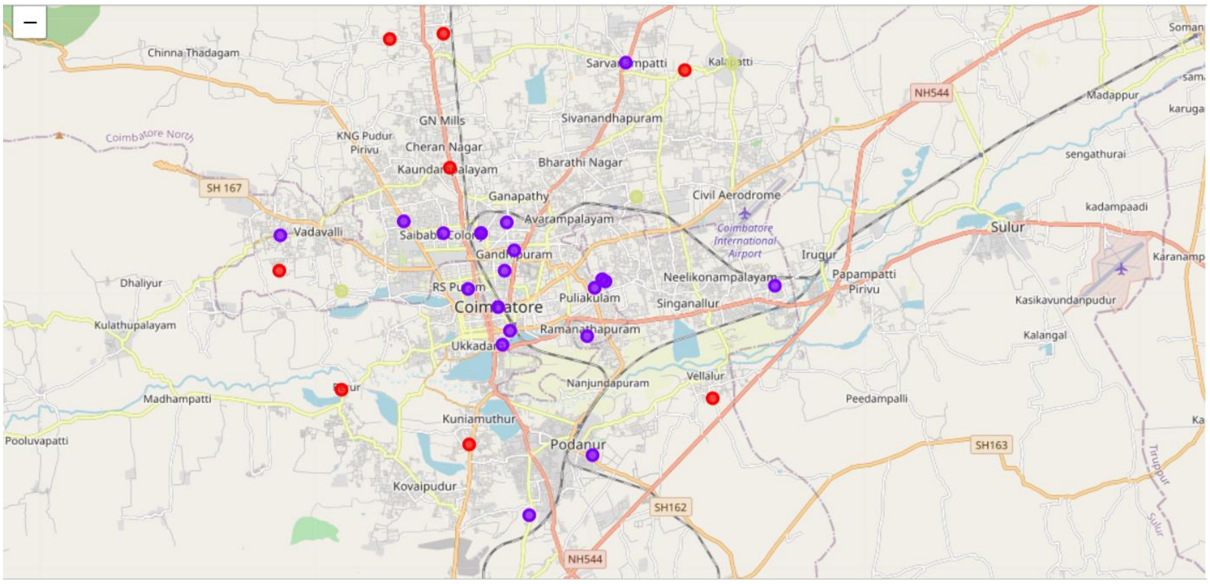
Results

The above mentioned, K-Means clustering method was applied to the dataframe of neighbourhoods of Coimbatore city. As mentioned earlier the number of clusters that was derived from elbow method was 4. The code as well as plotting of clusters can be seen below



In the next step we will visualize the clusters using Folium.

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

# Setup color scheme for different clusters
x = np.arange(cov_clusters)
ys = [i + x + (i*x)**2 for i in range(cov_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(cov_final['Latitude'], cov_final['Longitude'], cov_final['Neighbourhood'],
                                  cov_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
```



Discussion

As mentioned earlier the most suitable neighbourhoods for starting the restaurant business are present in the cluster number 1. Our K-Means model worked perfectly and successfully clustered similar neighbourhoods together. After studying all four clusters, it is recommended to the client that neighbourhoods such as **Ganapathy, Perur, Vilankurichi** that fall in **Cluster 1** look like good locations for starting their restaurant 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.

4. 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 analyse a geographical location because these libraries make it very easy to extract data that is easily available online. In this project we studied the neighbourhoods of Coimbatore city and came up with a recommendation of neighbourhoods where our client can start their restaurant business.