IBM Data Science Capstone

Analyzing Neighborhoods in Bengaluru, India to open a Shopping Mall

Sathvik Prabhu

July 2021

Introduction:

The urban population loves visiting shopping malls as it is a great way to relax and enjoy themselves during weekends and holidays. They can go grocery shopping, dine at restaurants, shop at the various fashion outlets, watch movies, and engage in arcade games and many more. Shopping malls are like a one-stop destination for all shopping needs. For retailers, the central location and the large crowd at the shopping malls provides a great distribution channel to market their products and services. Property developers are also taking advantage of this trend to build more shopping malls to cater to the demand.Opening shopping malls allows property developers to earn consistent rental income.
As a result, there are many shopping malls in the city of Bengaluru and many more are being built. Bengaluru also known as "*The Silicon Valley of India*" is the IT hub of India. It is the second fastest-growing major metropolis in India. Bangalore is a vibrant city which is always up and alive with its streets packed with people from all backgrounds. Of course, as with any business decision, opening a new shopping mall requires serious consideration and is a lot more complicated than it seems. Particularly, the location of the shopping mall is one of the most important decisions that will determine whether the mall will be a success or a failure.

Business Problem:

The objective of this capstone project is to analyse and select the best locations in the city of Bengaluru, India to open a new shopping mall. Using data science methodology and machine learning techniques like clustering, this project aims to provide solutions to answer the question : Which is the best location to open a shopping mall in Bengaluru, India?

Target Audience of this project:

This project is particularly useful to property developers and investors looking to open or invest in new shopping malls in Bengaluru , India.

Methodology:

The model has been created using python. Initially, the following packages have been imported:

```
import pandas as pd
import requests
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.cm as cm
import matplotlib.colors as colors
from geopy.geocoders import Nominatim
import geocoder
from pandas.io.json import json_normalize
import folium
```

Package Breakdown:

- pandas: To collect and manipulate data in JSON & HTML formats, and then data analysis
- Requests: Handle http requests
- matplotlib: Detailing the generated maps
- folium: Generating maps of Bengaluru
- sklearn: To import kmeans which is the machine learning model implemented.
- nominatim: Tool to search OpenStreetMap data by name and address.
- geocoder: To find coordinates of the neighborhoods in bangalore

The approach taken here is to explore the city, plot the map to show the neighbourhoods in consideration and then build the model by clustering all similar neighborhoods together and finally plot the new map with clustered neighborhoods. Insights are drawn and the findings are then discussed.

Data Collection :

The data of the neighborhoods in Bengaluru was scraped from https://en.wikipedia.org/wiki/List_of_neighbourhoods_in_Bangalore .

```
url = "https://en.wikipedia.org/wiki/List_of_neighbourhoods_in_Bangalore"
html_data = requests.get(url).text
```

```
temp_data = pd.read_html(html_data)
```

```
blr_data = pd.DataFrame()
for i in range (0,8):
    blr_data = pd.concat([blr_data, temp_data[i]], ignore_index=True)
blr_data
```

	Name	Image	Summary
0	Cantonment area	NaN	The Cantonment area in Bangalore was used as a...
1	Domlur	NaN	Formerly part of the Cantonment area, Domlur h...
2	Indiranagar	NaN	Indiranagar is a sought-after residential and ...
3	Rajajinagar	NaN	Established in 1949 on the birthday of C. Raja...
4	Malleswaram	NaN	NaN
...	...	...	...
60	Nandini Layout	NaN	NaN
61	Nayandahalli	NaN	Nayandahalli is a transport junction in the we...
62	Rajajinagar	NaN	NaN
63	Rajarajeshwari Nagar	NaN	Located in the south-western part of the city ...
64	Vijayanagar	NaN	Named after the Vijayanagara Empire, Vijayanag...

65 rows × 3 columns

The data is read into a pandas data frame using the read_html() method. This is done so that the Wikipedia page provides a comprehensive and detailed table of the data which can easily be scraped using the read_html() method of pandas.

Data Preprocessing:

The columns Image & Summary are irrelevant to the project and are hence dropped. The Cantonment Area is renamed as Bangalore Cantonment since the geocoder would then provide coordinates for cantonment areas outside bangalore. The column name is changed to ‘Neighborhood’ for the sake of simplicity.

```
blr_data.drop(['Image', 'Summary'], axis=1, inplace=True)
blr_data.rename(columns={'Name':"Neighborhood"}, inplace=True)
blr_data.at[0,'Neighborhood'] = "Bangalore Cantonment"
blr_data
```

	Neighborhood
0	Bangalore Cantonment
1	Domlur
2	Indiranagar
3	Rajajinagar
4	Malleswaram
...	...
60	Nandini Layout
61	Nayandahalli
62	Rajajinagar
63	Rajarajeshwari Nagar
64	Vijayanagar

65 rows × 1 columns

The resulting dataframe then looks like above.

Feature Engineering:

The geographical coordinates for Bengaluru, has been obtained from the geocoders library in python. This data is relevant for plotting the map of Bengaluru using the Folium library in python. The geocoder library in python has been used to obtain latitude and longitude data for various neighborhoods in Bengaluru. These coordinates are then further used for plotting using the Folium library in python.

```
# define a function to get coordinates
def get_latlng(neighborhood):
    # initializing variable to None
    lat_lng_coords = None
    while(lat_lng_coords is None):
        g = geocoder.arcgis('{}, Bangalore, India'.format(neighborhood))
        lat_lng_coords = g.latlng
    return lat_lng_coords
```

```
coords = [ get_latlng(neighborhood) for neighborhood in blr_data["Neighborhood"].tolist() ]
coords
```

...

```
df_coords = pd.DataFrame(coords, columns=['Latitude', 'Longitude'])
blr_data['Latitude'] = df_coords['Latitude']
blr_data['Longitude'] = df_coords['Longitude']
blr_data
```

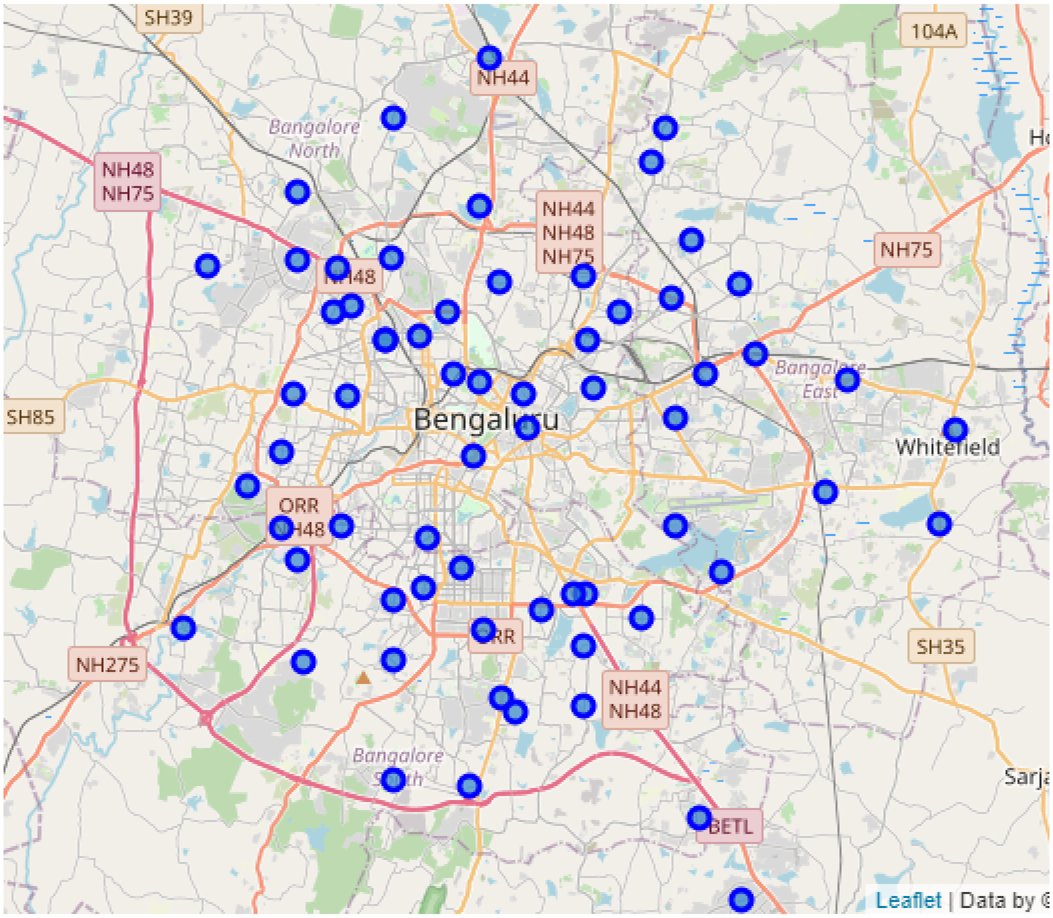
	Neighborhood	Latitude	Longitude
0	Bangalore Cantonment	12.97566	77.60542
1	Domlur	12.94329	77.65602
2	Indiranagar	13.03006	77.49526
3	Rajajinagar	13.00544	77.55693
4	Malleswaram	13.00632	77.56840
...	...	...	...
60	Nandini Layout	13.01481	77.53891
61	Nayandahalli	12.94205	77.52100
62	Rajajinagar	13.00544	77.55693
63	Rajarajeshwari Nagar	12.93178	77.52668
64	Vijayanagar	13.07600	77.65240

65 rows × 3 columns

The resulting dataframe after adding coordinates looks like above.

Visualizing the Neighbourhoods of Bengaluru:

Using the folium package, the above resulting dataframe is then used to visualize the map of Bengaluru.
Neighborhood map of Bengaluru:



Then using foursquare, we define a function which collects information pertaining to each neighbourhood including that of the name of the neighborhood, geo-coordinates, venue and venue categories.
The resulting data looks like this:

	Neighborhood	Latitude	Longitude	VenueName	VenueLatitude	VenueLongitude	VenueCategory
0	Bangalore Cantonment	12.97566	77.60542	M.G Road Boulevard	12.975771	77.603979	Plaza
1	Bangalore Cantonment	12.97566	77.60542	Blossom Book House	12.975042	77.604813	Bookstore
2	Bangalore Cantonment	12.97566	77.60542	Hysteria	12.974843	77.605426	Music Store
3	Bangalore Cantonment	12.97566	77.60542	Coast 2 Coast	12.975305	77.605625	Indian Restaurant
4	Bangalore Cantonment	12.97566	77.60542	The 13th Floor	12.975364	77.604995	Lounge

One Hot Encoding:

Label Encoding might cause the machine learning model to have a bias which is undesirable. To avoid this, One Hot Encoding is used. This helps to convert categorical data into numeric data. One hot encoding is performed and the mean of the grouped venue categories for each of the neighbourhoods is calculated.

```
# one hot encoding
blr_onehot = pd.get_dummies(venues_df[['VenueCategory']], prefix="", prefix_sep="")

# add neighborhood column back to dataframe
blr_onehot['Neighborhoods'] = venues_df['Neighborhood']

# move neighborhood column to the first column
fixed_columns = [blr_onehot.columns[-1]] + list(blr_onehot.columns[:-1])
blr_onehot = blr_onehot[fixed_columns]

print(blr_onehot.shape)
blr_onehot.head()
```

...

Grouping rows by neighborhood and by taking the mean of the frequency of occurence of each category

```
blr_grouped = blr_onehot.groupby(["Neighborhoods"]).mean().reset_index()

print(blr_grouped.shape)
blr_grouped
```

```
(64, 213)
```

	Neighborhoods	Afghan Restaurant	Airport	American Restaurant	Andhra Restaurant	Arcade	Art Gallery	Art Museum	Arts & Crafts Store	Asian Restaurant	...	Toy / Game Store
0	Anjanapura	0.0	0.0	0.000000	0.000000	0.00	0.0	0.0	0.000000	0.000000	...	0.0
1	Arekere	0.0	0.0	0.012195	0.000000	0.00	0.0	0.0	0.000000	0.000000	...	0.0
2	BTM Layout	0.0	0.0	0.000000	0.010989	0.00	0.0	0.0	0.000000	0.010989	...	0.0
3	Banashankari	0.0	0.0	0.000000	0.000000	0.01	0.0	0.0	0.000000	0.020000	...	0.0
4	Banaswadi	0.0	0.0	0.000000	0.017857	0.00	0.0	0.0	0.017857	0.017857	...	0.0

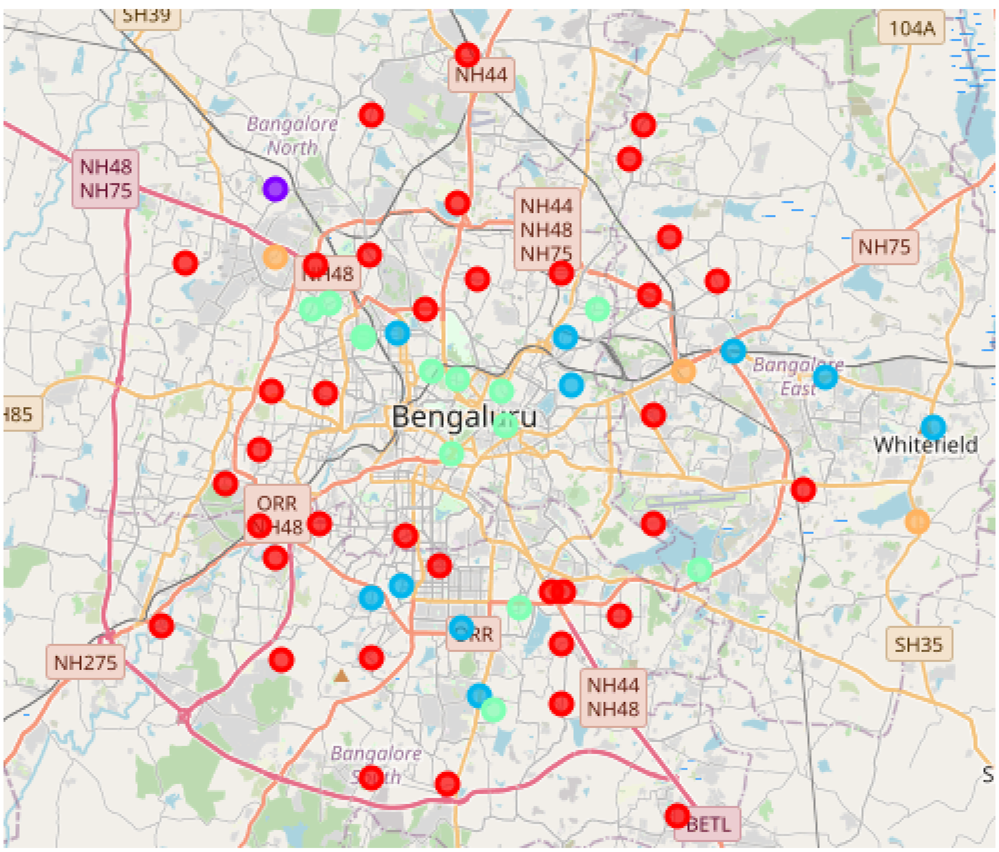
Model Building - KMeans:

Using KMeans clustering machine learning algorithm, similar neighborhoods are clustered together. Each of the neighborhoods are labelled and the label is added into the dataset. The resulting dataframe looks like this:

	Neighborhood	Shopping Mall	Cluster Labels	Latitude	Longitude
0	Anjanapura	0.000000	0	12.85811	77.55910
27	Kalyan Nagar	0.000000	0	12.96802	77.52114
28	Kamakshipalya	0.000000	0	12.98699	77.52484
30	Kengeri	0.000000	0	12.90868	77.48718
31	Koramangala	0.000000	0	12.92004	77.62546
...	...	...	...	...	...
5	Bangalore Cantonment	0.010000	3	12.97566	77.60542
43	Nandini Layout	0.012500	3	13.01481	77.53891
46	Peenya	0.043478	4	13.03188	77.52654
57	Varthur	0.033333	4	12.94349	77.74701
37	Mahadevapura	0.043478	4	12.99409	77.66633

Visualizing the clustered neighborhoods:

The data is processed, missing data is collected and compiled. The model is built. Now, the clustered neighborhoods are visualized on the map using the Folium Package.
Map of Clustered Neighborhoods in Bengaluru:



Examining the clusters:

The clusters are then examined by expanding the code using the cluster labels column.

Cluster 1

```
blr_merged.loc[blr_merged['Cluster Labels'] == 0]
```

Cluster 2

```
blr_merged.loc[blr_merged['Cluster Labels'] == 1]
```

Cluster 3

```
blr_merged.loc[blr_merged['Cluster Labels'] == 2]
```

Cluster 4

```
blr_merged.loc[blr_merged['Cluster Labels'] == 3]
```

Cluster 5

```
blr_merged.loc[blr_merged['Cluster Labels'] == 4]
```

Discussion:

The purpose of this project was to analyze neighborhoods in Bengaluru, India to open a shopping mall. It can be observed that most of the shopping malls are concentrated in the northern and eastern areas of Bengaluru, with the highest number in cluster 2 and moderate number in cluster 5 as well as cluster 3. Cluster 1 has little to no number of malls in its neighborhoods. This is a great opportunity and serves as a high potential area to open new shopping malls as there is hardly any competition from existing malls. Meanwhile, shopping malls in clusters 1 and 5 have high competition and therefore it's advisable to avoid these neighborhoods to invest or build new shopping malls. This project thereby recommends property developers to capitalize on these findings to open new shopping malls in neighborhoods in cluster 1. Property Developers with unique selling propositions can also open new shopping malls in neighborhoods in cluster 3 & 4 with moderate competition. Lastly, property developers are advised to avoid neighborhoods in cluster 2 and cluster 5 which already have high concentration of shopping malls and are suffering from intense competition

Conclusion:

In this project, the neighborhoods in Bengaluru, India have been successfully analyzed for determining which would be the best neighborhoods for opening a new shopping mall. Based on the analysis carried out, neighborhoods in cluster 1 are recommended as locations for the new mall. These neighborhoods are plotted and visualized as shown above. The stakeholders and investors can further tune this by considering various other factors like transport, legal requirements, and costs associated. These were out of the scope for this project and thus were not considered.