Coursera Capstone
IBM Applied Data Science Capstone

Opening a New Shopping Mall in Bangalore, India

By
Dipankar Roy

May 2020

Introduction

Bangalore is a vibrant city rich with its multilinguistic art and culture. The city cosmopolitan culture and trendy lifestyle have its charm and allure which has attracted a lot of people from different parts of the county. With good no places of entertainment and be a stress buster during your weekends, Bangalore has a lot to offer.

Its busy streets with people and the craze about shopping among the people have contributed to a number of shopping malls in Bangalore. So, for a shopaholic and the people who want to explore a lot of things under one roof then what can be more exciting than visiting a shopping mall.

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. As a result, there are many Shopping Malls in the city of Bangalore. Opening Shopping Malls allows property developers to earn consistent rental income. 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 project is to analyse and select the best locations in the city of Bangalore, 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 business question: In the city of Bangalore, India, if a property developer is looking forward to open a new Shopping Mall, where would you recommend that they open it?

Data

To solve the problem, the following data is required:

- List of neighbourhoods in Bangalore. This defines the scope of this project which is confined to the city of Bangalore.
- Latitude and longitude coordinates of those neighbourhoods. This is required in order to plot the map and also to get the venue data.
- Venue data, particularly data related to Shopping Malls. This data will be used to perform clustering on the neighbourhoods.

Sources of data and methods to extract them

This Wikipedia page https://en.wikipedia.org/wiki/Category:Neighbourhoods_in_Bangalore contains a list of neighbourhoods in Bangalore, with a total of 128 neighbourhoods. Web scraping techniques were used to extract the data from the Wikipedia page, with the help of Python requests and beautifulsoup packages. Geographical coordinates of the neighbourhoods were found using Python Geocoder package which gives us the latitude and longitude of the neighbourhoods.

Foursquare API were used to get the venue data for those neighbourhoods based on their coordinates. Foursquare has one of the largest database of 105+ million places and is used by over 125,000 developers.

Foursquare API will provide many categories of the venue data, we are particularly interested in the Shopping Mall category in order to help us to solve the business problem put forward.

This is a project that will make use of many data science skills, from web scraping (Wikipedia), working with API (Foursquare), data cleaning, data wrangling, to machine learning (K-means clustering) and map visualization (Folium). In the next section, we will present the Methodology section where we will discuss the steps taken in this project, the data analysis that we did and the machine learning technique that was used.

Methodology

The list of neighbourhoods in the city of Bangalore is available in the Wikipedia page https://en.wikipedia.org/wiki/Category:Neighbourhoods_in_Bangalore. The list of neighbourhoods data was extracted by web scraping using Python requests and beautifulsoup packages .

```
# send the GET request
data = requests.get("https://en.wikipedia.org/wiki/
   Category:Neighbourhoods_in_Bangalore").text
# parse data from the html into a beautifulsoup object
soup = BeautifulSoup(data, 'html.parser')
# create a list to store neighborhood data
neighbourhoodList = []
# append the data into the list
for row in soup.find_all("div", class_="mw-category")[0].
   findAll("li"):
    neighbourhoodList.append(row.text)
# create a new DataFrame from the list
blr_df = pd.DataFrame({"Neighbourhood": neighbourhoodList
   })
```

However, this is just a list of names. We need to get the geographical coordinates in the form of latitude and longitude in order to be able to use Foursquare API. To do so, the Geocoder package was used that allowed us to convert address into geographical coordinates in the form of latitude and longitude.

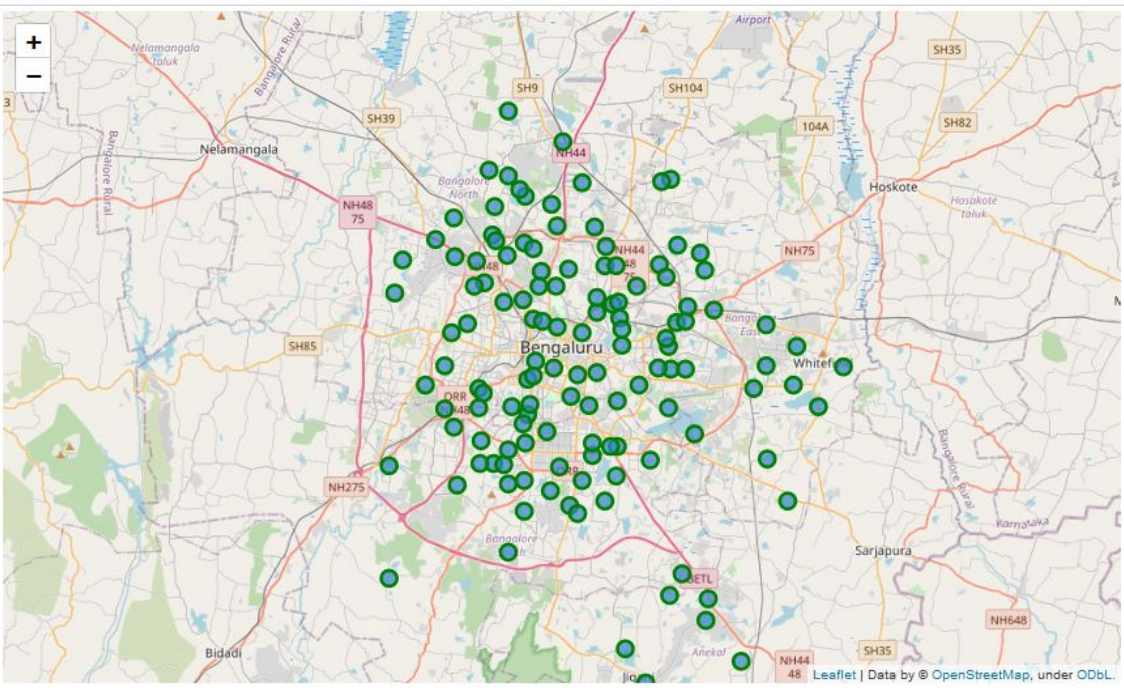
```
# define a function to get coordinates
def get_latlng(neighbourhood):
    # initialize your variable to None
    lat_lng_coords = None
    # loop until you get the coordinates
    while(lat_lng_coords is None):
        g = geocoder.arcgis('{}, Bangalore, India'.format
            (neighbourhood))
        lat_lng_coords = g.latlng
    return lat_lng_coords
# call the function to get the coordinates, store in a
   new list using list comprehension
coords = [ get_latlng(neighborhood) for neighborhood in
   blr_df["Neighbourhood"].tolist() ]
```

After gathering the data, the data was populated into a pandas DataFrame and then the neighbourhoods were visualized in a map using Folium package. This allowed to perform a sanity check to make sure that the geographical coordinates data returned by Geocoder are correctly plotted in the city of Bangalore.

```
# get the coordinates of Bangalore
address = 'Bangalore, India'

geolocator = Nominatim(user_agent="my-application")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Bangalore, India {},
    {}.'.format(latitude, longitude))
# create map of Bangalore using latitude and longitude
   values
map_blr = folium.Map(location=[latitude, longitude],
   zoom_start=11)
# add markers to map
for lat, lng, neighbourhood in zip(blr_df['Latitude'],
   blr_df['Longitude'], blr_df['Neighbourhood']):
    label = '{}'.format(neighbourhood)
    label = folium.Popup(label, parse_html=True)
    folium.CircleMarker(
        [lat, lng],
        radius=7,
        popup=label,
```

```
        color='green',
        fill=True,
        fill_color='#3186cc',
        fill_opacity=0.7).add_to(map_blr)

map_blr
```

Output:



Next, the Foursquare API was used to get the top 100 venues that are within a radius of 2000 meters. API calls were made to Foursquare passing in the geographical coordinates of the neighbourhoods in a Python loop. Foursquare returnued the venue data in JSON format and the venue name, venue category, venue latitude and longitude were extracted. With the data, number of venues returned were checked for each neighbourhood and it was also examined how many unique categories could be curated from all the returned venues. Then, each neighbourhood were analysed by grouping the rows by neighbourhood and taking the mean of the frequency of occurrence of each venue category. By doing so, the data was also prepared for use in clustering. Since we are analysed the "Shopping Mall" data, "Shopping Mall" filted was used as venue category for the neighbourhoods.

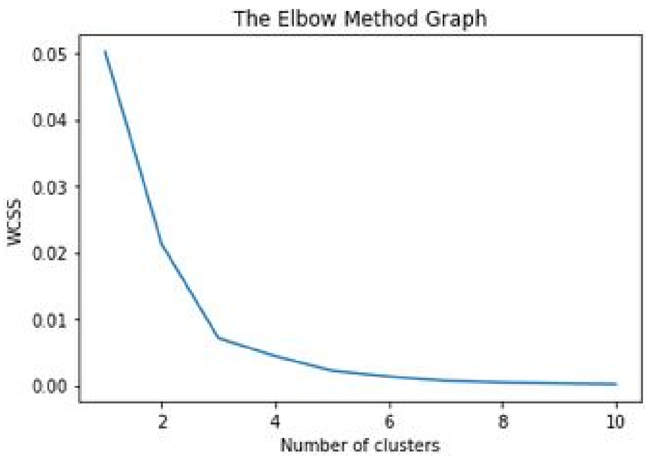
```
# define Foursquare Credentials and Version
CLIENT_ID = '****'
CLIENT_SECRET = '***'
VERSION = '20180605' # Foursquare API version

print('Your credentails:')
print('CLIENT_ID: ' + CLIENT_ID)
```

```python
print('CLIENT_SECRET:' + CLIENT_SECRET)
radius = 2000
LIMIT = 100

venues = []

for lat, long, neighborhood in zip(blr_df['Latitude'], blr_df['Longitude'], blr_df['Neighbourhood']):

    # create the API request URL
    url = "https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&v={}&ll={},{}&radius={}&limit={}".format(
        CLIENT_ID,
        CLIENT_SECRET,
        VERSION,
        lat,
        long,
        radius,
        LIMIT)

    # make the GET request
    results = requests.get(url).json()["response"]['groups'][0]['items']

    # return only relevant information for each nearby venue
    for venue in results:
        venues.append((
            neighborhood,
            lat,
            long,
            venue['venue']['name'],
            venue['venue']['location']['lat'],
            venue['venue']['location']['lng'],
            venue['venue']['categories'][0]['name']))
# convert the venues list into a new DataFrame
venues_df = pd.DataFrame(venues)

# define the column names
venues_df.columns = ['Neighbourhood', 'Latitude', 'Longitude', 'VenueName', 'VenueLatitude', 'VenueLongitude', 'VenueCategory']

print(venues_df.shape)
venues_df.head()
```

Lastly, clustering was performed on the data by using k-means clustering. K-means clustering algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible. It is one of the simplest and popular unsupervised machine learning algorithms and is particularly suited to solve the problem for this project. Elbow method was used to find the best K value.

```
import matplotlib.pyplot as plt
%matplotlib inline
wcss=[]
#kl_mall.drop(["Neighbourhood"], 1)
for i in range(1,11):
    kmeans = KMeans(n_clusters=i, init ='k-means++',
       max_iter=300,  n_init=10,random_state=0 )
    kmeans.fit(blr_mall)
    wcss.append(kmeans.inertia_)

plt.plot(range(1,11),wcss)
plt.title('The Elbow Method Graph')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

Output:



from the above graph, best number of clusters was found to be 3 based on their frequency of occurrence for "Shopping Mall". The results allowed us to identify which neighbourhoods have higher concentration of Shopping Malls while which neighbourhoods have fewer number of Shopping Malls. Based on the occurrence of Shopping Malls in different neighbourhoods. the data would help us to answer the question as to which neighbourhoods are most suitable to open new Shopping Malls.

```
# set number of clusters
kclusters = 3

```

```
# run k-means clustering
kmeans = KMeans(n_clusters=kclusters, random_state=0).fit
    (blr_mall)

# check cluster labels generated for each row in the
    dataframe
kmeans.labels_[0:100]
# create a new dataframe
blr_merged = blr_mall_copy.copy()
# add clustering labels
blr_merged["Cluster Labels"] = kmeans.labels_
```

Results

The results from the k-means clustering show that we can categorize the neighbourhoods into 3 clusters based on the frequency of occurrence for "Shopping Mall":

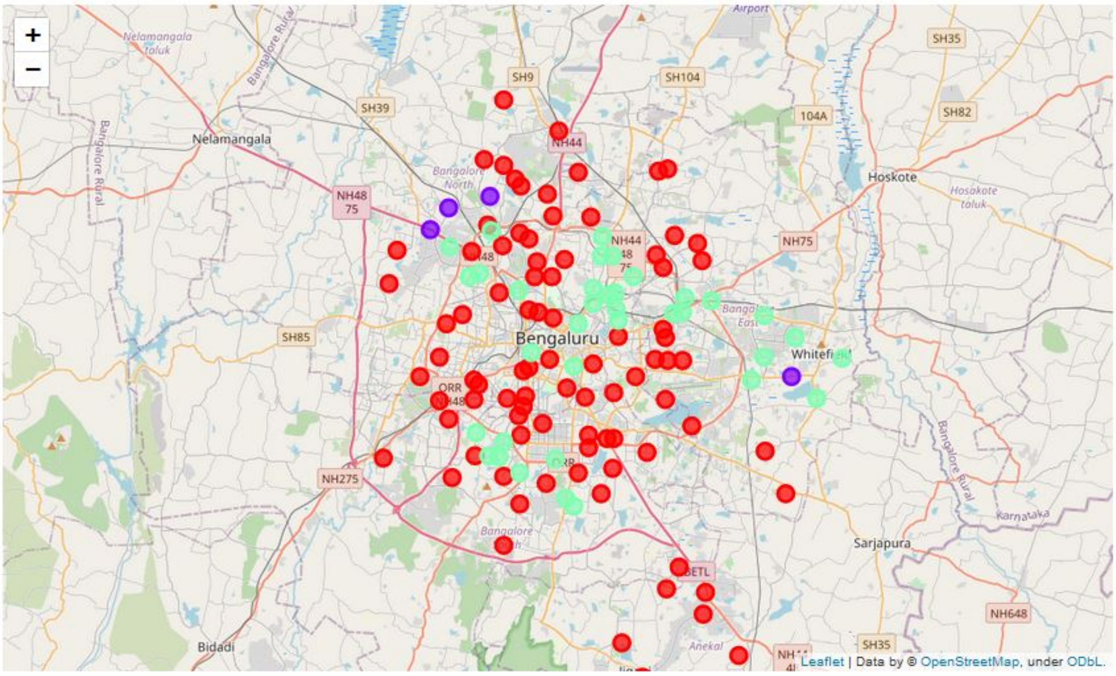
- Cluster 0: Neighbourhoods with moderate number of Shopping Malls
- Cluster 1: Neighbourhoods with low number to no existence of Shopping Malls
- Cluster 2: Neighbourhoods with high concentration of Shopping Malls

Visualization of results were don using the following code:

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

# 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(blr_merged['Latitude'],
    blr_merged['Longitude'], blr_merged['Neighbourhood'],
    blr_merged['Cluster Labels']):
    label = folium.Popup(str(poi) + ' - Cluster ' + str(
        cluster), parse_html=True)
    folium.CircleMarker(
        [lat, lon],
        radius=7,
        popup=label,
        color=rainbow[cluster-1],
```

```
        fill=True,
        fill_color=rainbow[cluster-1],
        fill_opacity=0.7).add_to(map_clusters)

map_clusters
```

The results of the clustering are visualized in the map below with cluster 0 in red colour, cluster 1 in purple colour, and cluster 2 in mint green colour.



Discussion

As observations noted from the map in the Results section, most of the Shopping Malls are concentrated in the central area of Bangalore city, with the highest number in cluster 2 and moderate number in cluster 0. On the other hand, cluster 1 has very low number to no Shopping Mall in the neighbourhoods. This represents a great opportunity and high potential areas to open new Shopping Malls as there is very little to no competition from existing malls. Meanwhile, Shopping Malls in cluster 2 are likely suffering from intense competition due to oversupply and high concentration of Shopping Malls. From another perspective, the results also show that the oversupply of Shopping Malls mostly happened in the central area of the city, with the suburb area still have very few Shopping Malls. Therefore, this project recommends property developers to capitalize on these findings to open new Shopping Malls in neighbourhoods in cluster 1 with little to no competition. Property developers with unique selling propositions to stand out from the competition can also open new Shopping Malls in neighbourhoods in cluster 0 with moderate competition. Lastly, property developers are advised to avoid neighbourhoods in cluster 2 which already have high concentration of Shopping Malls and suffering from intense competition.

Conclusion

In this project, we have gone through the process of identifying the business problem, specifying the data required, extracting and preparing the data, performing machine learning by clustering the data into 3 clusters based on their similarities, and lastly providing recommendations to the relevant stakeholders i.e. property developers and investors regarding the best locations to open a new Shopping Mall. To answer the business question that was raised in the introduction section, the answer proposed by this project is: The neighbourhoods in cluster 1 are the most preferred locations to open a new Shopping Mall. The findings of this project will help the relevant stakeholders to capitalize on the opportunities on high potential locations while avoiding overcrowded areas in their decisions to open a new Shopping Mall.

References

Category:Suburbs in Bangalore. Wikipedia. Retrieved from
https://en.wikipedia.org/wiki/Category:Neighbourhoods_in_Bangalore
Foursquare Developers Documentation. Foursquare. Retrieved from
https://developer.foursquare.com/docs