Identifying suitable location to start a restaurant

Business Problem:

I want to start a restaurant business in Madhapur location of Hyderabad city.Identifying a proper location for restaurant is a big task.The location is an essential factor to consider while discussing how to start a restaurant business, as it can determine the success of your restaurant. While choosing your restaurant's location, it is a good idea to identify your competitor in that area and gauge their progress and understand their business model.The restaurant should be located in a place that is easily visible as well as accessible.

The size of the location is an essential factor to be considered during the restaurant site selection process. The size of the restaurant does not just imply the dining space. The site should be big enough to accommodate all the kitchen equipment. Walk-in freezer, cooking stations, etc. all require large spaces. Depending on your concept you can decide the size of the seating area, bar, etc. A cramped up space to accommodate more customers often works the opposite and drives customers away because of the small and narrow dining space.

Data:

To solve the problem, we will need the following data: • List of neighbourhoods in Hyderabad. This defines the scope of this project which is confined to the city of Hyderabad, the capital city of Telangana, which is in South India • 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 Restaurants. We will use this data to perform clustering on the neighbourhoods.

Foursquare API will provide many categories of the venue data, and we are particularly interested in the Restaurants category in order to help us solve the business problem. 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).

Sources of Data and methods to extract the Data:

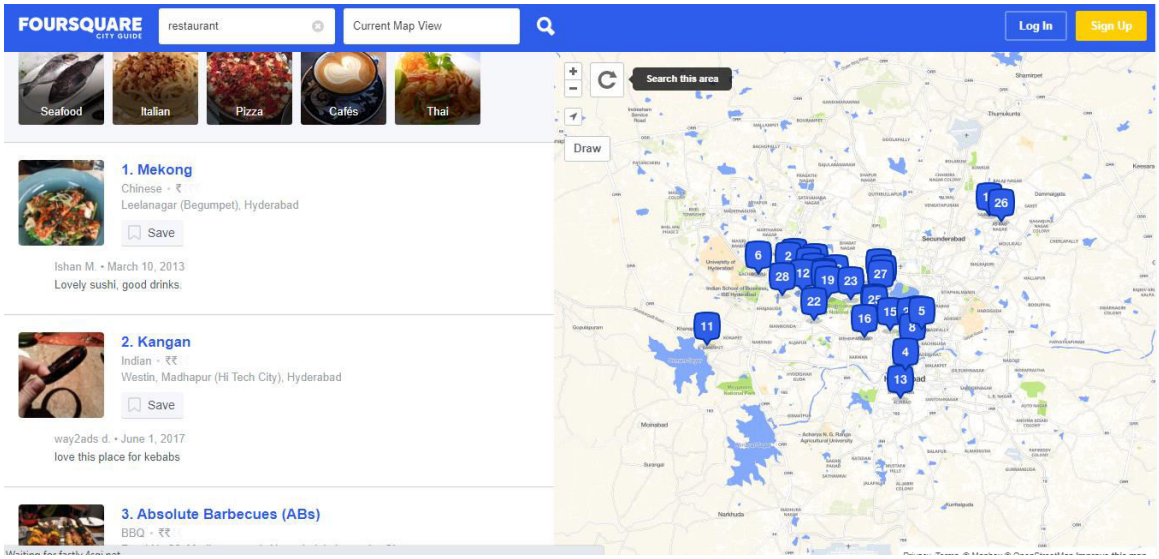
This https://en.wikipedia.org/wiki/Category:Neighbourhoods_in_Hyderabad,_India is a list of neighbourhoods in Hyderabad, with 200 neighbourhoods. I have used web scraping techniques to extract the data from the Wikipedia page, with the help of Python requests and beautifulsoup packages. Then we can get the latitude and longitude coordinates of the neighbourhoods using Python Geocoder package. After that, I have used the Foursquare API to get the venue data for

those neighbourhoods.

Foursquare API will provide many categories of the venue data, and we are particularly interested in the Restaurant category in order to help us solve the business problem. 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).

Methodology

The Foursquare API allows application developers to interact with the Foursquare platform. The API itself is a RESTful set of addresses to which you can send requests, so there's really nothing to download onto your server.



Web Scraping

Perform scraping using Python requests and beautifulsoup packages to extract the list of neighbourhoods data.

```
# Send the GET request

data =
requests.get("https://en.wikipedia.org/wiki/Category:Neighbourhoods_in_Hyderabad,_India").text

# Parse data from the html into a beautifulsoup object

soup = BeautifulSoup(data, 'html.parser')

# Create a list to store neighbourhood data

neighborhoodList = []

# Append the data into the list

for row in soup.find_all("div", class_="mw-category")
[0].findAll("li"):
  neighborhoodList.append(row.text)

# Create a new DataFrame from the list

kl_df = pd.DataFrame({"Neighborhood": neighborhoodList})
kl_df.head()
```

	Neighborhood
0	A. S. Rao Nagar
1	A.C. Guards
2	Abhyudaya Nagar
3	Abids
4	Adikmet
5	Afzal Gunj
6	Aghapura
7	Aliabad, Hyderabad
8	Alijah Kotla
9	Allwyn Colony
10	Alwal
11	Amberpet

This is the data frame created after scraping the data. 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, we will use the Geocoder package that will allow us to convert the address into geographical coordinates in the form of latitude and longitude.

```
# Defining a function to get coordinates

def get_latlng(neighborhood):
    # initialize your variable to None
    lat_lng_coords = None
    # loop until you get the coordinates
    while(lat_lng_coords is None):
        g = geocoder.arcgis('{}, Hyderabad,
India'.format(neighborhood))
        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
kl_df["Neighborhood"].tolist()]
```

```
coords
```

```
[[17.411200000000065, 78.50824000000006],
 [17.393000949133675, 78.45689980427697],
 [17.337650000000053, 78.56414000000007],
 [17.389800000000037, 78.47658000000007],
 [17.410610000000077, 78.51513000000006],
 [17.37751000000003, 78.48005000000006],
 [17.38738496982723, 78.46699458034638],
 [17.34259000000003, 78.47626000000008],
 [17.36068000000006, 78.47998000000007],
 [17.503370000000075, 78.41602000000006],
 [17.535430000000076, 78.54427000000004],
 [17.385820000000024, 78.51836000000003],
 [17.53332000000006, 78.32529000000005],
 [17.435350000000028, 78.44861000000003],
 [17.45787000000007, 78.53882000000004],
 [17.40784000000002, 78.49150000000003],
 [17.385140000000035, 78.44738000000007],
 [17.369170000000054, 78.43683000000004],
 [17.40710000000007, 78.50233000000003],
 [17.372720000000072, 78.49047000000007],
```

We have obtained the latitude and longitude coordinates for all the places, so we need to merge the coordinates into the original data frame.

```
# Create temporary dataframe to populate the coordinates into
Latitude and Longitude

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

# Merge the coordinates into the original dataframe

kl_df['Latitude'] = df_coords['Latitude']
kl_df['Longitude'] = df_coords['Longitude']

print(kl_df.shape)
kl_df

(200,3)
```

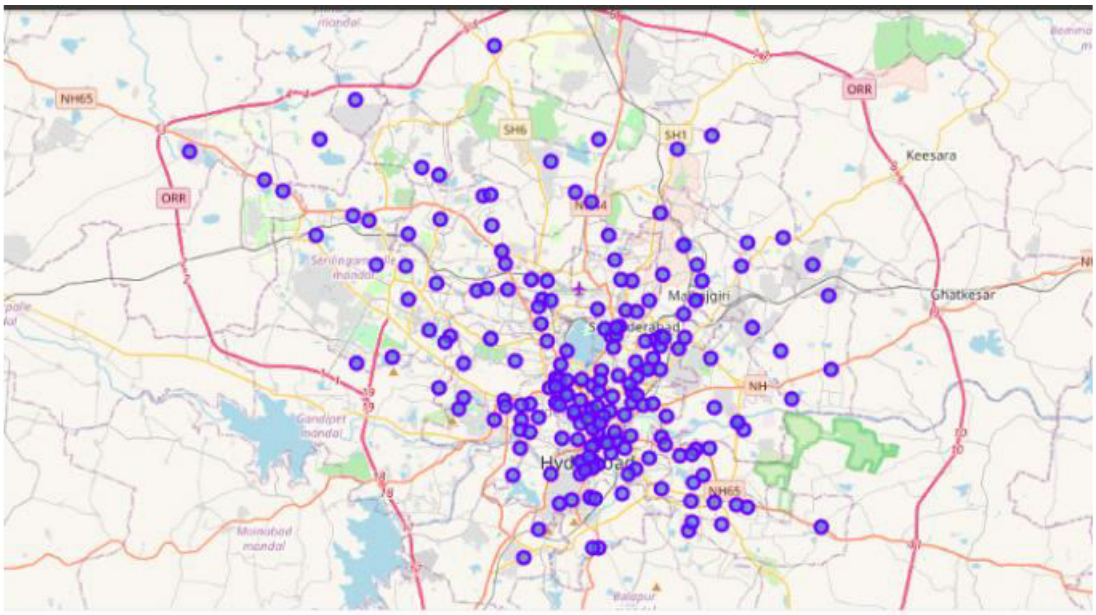
	Neighborhood	Latitude	Longitude
0	A. S. Rao Nagar	17.411200	78.508240
1	A.C. Guards	17.393001	78.456900
2	Abhyudaya Nagar	17.337650	78.564140
3	Abids	17.389800	78.476580
4	Adikmet	17.410610	78.515130
5	Afzal Gunj	17.377510	78.480050
6	Aghapura	17.387385	78.466995
7	Aliabad, Hyderabad	17.342590	78.476260
8	Alijah Kotla	17.360680	78.479980
9	Allwyn Colony	17.503370	78.416020
10	Alwal	17.535430	78.544270
11	Amberpet	17.385820	78.518360
12	Ameenpur	17.533320	78.325290
13	Ameerpet	17.435350	78.448610
14	Anandbagh	17.457870	78.538820
15	Ashok Nagar, Hyderabad	17.407840	78.491500

This is the combined data frame which contains all the neighbourhoods along with the geographical coordinates.

```
# Getting the coordinates of Hyderabad


address = 'Hyderabad, India'
geolocator = Nominatim(user_agent="my-application")
location = geolocator.geocode(address)
latitude = location.latitude
longitude = location.longitude
print('The geograpical coordinate of Hyderabad, India {},
{}.'.format(latitude, longitude))
```

After gathering the data, we have to populate the data into a pandas DataFrame and then visualize the neighbourhoods in a map using Folium package.

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

# Adding markers to map

for lat, lng, neighborhood in zip(kl_df['Latitude'],
kl_df['Longitude'], kl_df['Neighborhood']):
 label = '{}'.format(neighborhood)
 label = folium.Popup(label, parse_html=True)
 folium.CircleMarker([lat,
lng],radius=5,popup=label,color='blue',fill=True,fill_color='#3186cc'
,fill_opacity=0.7).add_to(map_kl)
map_kl
```



Hyderabad map containing all the neighbourhoods

Use the foursquare API to explore the neighbourhoods:

```
CLIENT_ID = '' # your Foursquare ID
CLIENT_SECRET = '' # your Foursquare Secret
VERSION = '20180604'


radius = 2000
LIMIT = 100
venues = []
for lat, long, neighborhood in zip(kl_df['Latitude'],
kl_df['Longitude'], kl_df['Neighborhood']):


# 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']))
```

After extracting all the venues, we have to convert the venues list into a new DataFrame.

```
CLIENT_ID = '' # your Foursquare ID
CLIENT_SECRET = '' # your Foursquare Secret
VERSION = '20180604'


radius = 2000
LIMIT = 100
venues = []
for lat, long, neighborhood in zip(kl_df['Latitude'],
kl_df['Longitude'], kl_df['Neighborhood']):


# 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']))
```

```
venues_df = pd.DataFrame(venues)
# Defining the column names


venues_df.columns = ['Neighborhood', 'Latitude', 'Longitude',
'VenueName', 'VenueLatitude', 'VenueLongitude', 'VenueCategory']
print(venues_df.shape)
venues_df.head()
```

	Neighborhood	Latitude	Longitude	VenueName	VenueLatitude	VenueLongitude	VenueCategory
0	A. S. Rao Nagar	17.4112	78.50824	Bawarchi	17.406369	78.497662	Indian Restaurant
1	A. S. Rao Nagar	17.4112	78.50824	Subway	17.404173	78.514950	Sandwich Place
2	A. S. Rao Nagar	17.4112	78.50824	Sudharshan Theatre 35mm	17.406530	78.495150	Movie Theater
3	A. S. Rao Nagar	17.4112	78.50824	Devi 70 MM	17.406329	78.495409	Movie Theater
4	A. S. Rao Nagar	17.4112	78.50824	Baskin-Robbins	17.404311	78.510034	Ice Cream Shop
5	A. S. Rao Nagar	17.4112	78.50824	Cafe Coffee Day	17.423051	78.501520	Coffee Shop
6	A. S. Rao Nagar	17.4112	78.50824	Spencer's	17.412592	78.498400	Convenience Store
7	A. S. Rao Nagar	17.4112	78.50824	Daily Bread	17.403554	78.514961	Café
8	A. S. Rao Nagar	17.4112	78.50824	Crystal Restaurant	17.406608	78.496268	Asian Restaurant
9	A. S. Rao Nagar	17.4112	78.50824	Sandhya 70 MM	17.407053	78.497724	Movie Theater

Analyzing each neighbourhood:

Here we apply one hot encoding to all the venues. So now the number of columns becomes 175

```
# One hot encoding

kl_onehot = pd.get_dummies(venues_df[['VenueCategory']], prefix="",
prefix_sep="")

# Adding neighborhood column back to dataframe

kl_onehot['Neighborhoods'] = venues_df['Neighborhood']

# Moving neighbourhood column to the first column

fixed_columns = [kl_onehot.columns[-1]] +
list(kl_onehot.columns[:-1])
kl_onehot = kl_onehot[fixed_columns]

print(kl_onehot.shape)

(6684, 175)
```

Clustering the neighbourhoods:

Now we need to cluster all the neighbourhoods into different clusters. The results will allow us to identify which neighbourhoods have a higher concentration of Restaurants while which neighbourhoods have a fewer number of Restaurants. Based on the occurrence of Restaurants in different neighbourhoods, it will help us answer the question as to which neighbourhoods are most suitable to open new Restaurants.

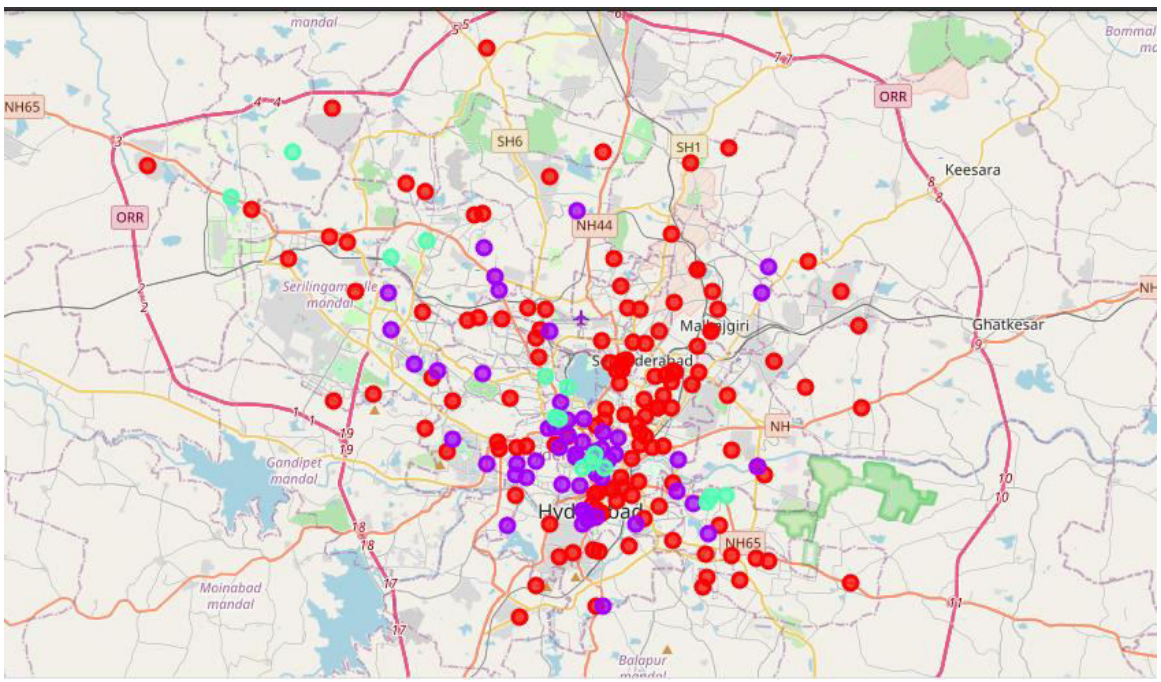
```
# Setting the number of clusters

kclusters = 1

kl_clustering = kl_mall.drop(["Neighborhoods"], 1)

# Run k-means clustering algorithm

kmeans = KMeans(n_clusters=kclusters,random_state=0).fit(kl_clustering)

# Checking cluster labels generated for each row in the dataframe

kmeans.labels_[0:10]
```

After applying the K-Means clustering algorithm, all the neighbourhoods get segregated and form different clusters.

```
# Creating a new dataframe that includes the cluster as well as the
top 10 venues for each neighborhood.


kl_merged = kl_mall.copy()


# Add the clustering labels


kl_merged["Cluster Labels"] = kmeans.labels_
kl_merged.rename(columns={"Neighborhoods": "Neighborhood"},
inplace=True)
kl_merged.head(10)
```

Visualizing the resulting clusters:

```
# Creating the map


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


# Setting 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(kl_merged['Latitude'],
kl_merged['Longitude'], kl_merged['Neighborhood'], kl_merged['Cluster
Labels']):


label = folium.Popup(str(poi) + ' - Cluster ' + str(cluster),
parse_html=True)

folium.CircleMarker([lat,lon],radius=5,popup=label,color=rainbow[clus
ter-1],fill=True,fill_color=rainbow[cluster-
1],fill_opacity=0.7).add_to(map_clusters)


map_clusters
```



Conclusions

A good number of Restaurants are concentrated in the central area of Hyderabad city. Cluster 0 has a very low number of Restaurants. This represents a great opportunity and high potential areas to open new Restaurants, as there is very little to no competition from existing Restaurants . Meanwhile, Restaurants in cluster 2 are likely suffering from intense competition because of oversupply and a high concentration of Restaurants. Therefore, this project recommends property developers to capitalize on these findings to open new Restaurants in neighbourhoods in cluster 0 with little to no competition. Property developers with unique selling propositions to stand out from the competition can also open new Restaurants in neighbourhoods in cluster 1 with moderate competition. Lastly, property developers are advised to avoid neighbourhoods in cluster 2 which already have a high concentration of Restaurants and suffering from intense competition.