Data tells where to build new hotels in Berlin

I. Introduction

1.1 Background

Berlin, the capital and the largest city of Germany, has always been among the top popular cities. The city attracts millions of tourists every year to explore their cultural richness. Apart from that, Berlin is also a major center of politics, media, and science. Therefore, the need for accommodation in this city is always on high demand.

An entrepreneur wants to invest in building up a chain of hotels in Berlin, Germany. He is new to this city, so he wants to find the best area in Berlin where he should set up the hotel chain. The common sense for solving this problem is to have the hotel nearest to the famous tourist attractions. Other suggestion could be close to main public stations. However, every city has their own hidden gems, where can attract more travelers than it looks like, which can be an extraordinary sight or a coffee shop. Let's take an example in Berlin. Being so well-known for the Brandenburg Gate or the Berlin Wall, but it should not deny that a Hatch Sticker Museum could also attract many people to come and visit the art and history of the graffiti sticker. Or perhaps the best place to try currywurst (curry sausage) in Berlin could be the small food stall on the street. The travelers who visited Berlin previously, are those who can tell us what the best places to be in Berlin are.

On the other hand, social media has become an integral part of all of our lives. People use social media to connect with friends, to share experience, or to catch up with trends. As such, tourists have also shared their experiences by providing likes, comments or ratings for the places they have visited. Thanks to that, business has an extremely important source of data to explore. Understanding that popular trend, we decide to use the data from people visiting Berlin on Foursquare to find where this entrepreneur should set up his hotel chain.

1.2. Problem

How to find the best possible places in Berlin to build up a new hotel chain in Berlin using the data retrieved from Foursquare?

1.3. Solution

So the proposed idea in order to solve this challenge for the entrepreneur is to build the ***hotel chain*** in the location that in the ***centroids*** of interesting places which receive ***high ratings*** from travelers on Foursquare. Since Berlin is a big city with the area of approximately 900 km2, we initially plan to have at least 5 hotels in the city. As such, we will use the *k-Clustering* to identify the 5 main clusters, and use the centroids of these clusters to suggest as the best place to build hotels

1.4. Assumption

To do that, several assumptions have been made for this project:

- A place is considered as interesting if the rating is 7.5 (out of 10) and above

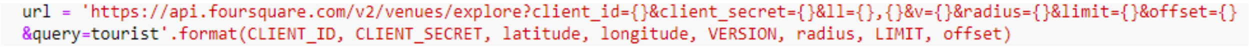
- The entrepreneur has sufficient capital, so we do not consider the impact of capital in decision making process
- Considering the area of Berlin is about 891.8km2, we make the assumption that the most interesting places should be within the radius of 10 km from Berlin city center
- With the sandbox account, we have the limit of the premium calls, so we are unfortunately unable to explore the full set of data.

II. Dataset preparation & description

The data required to solve the problem is the Foursquare data for interesting places in Berlin, Germany.

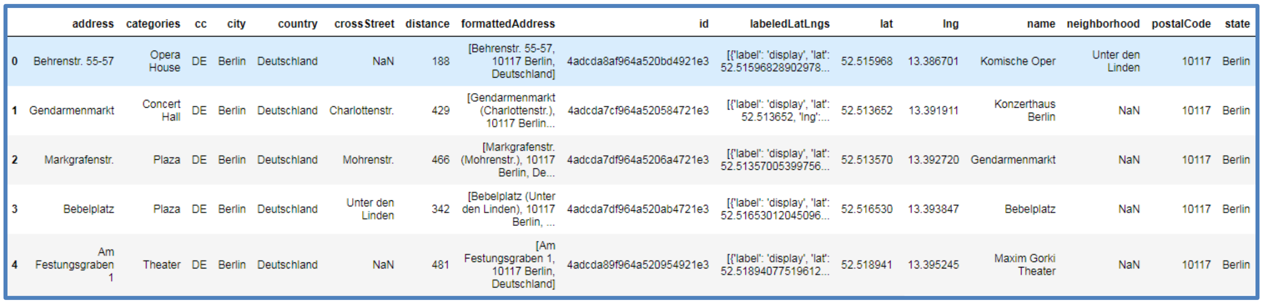
We first choose the city center as our starting point. From this point, we will explore venues within the radius as 10,000m (as *Assumption 3*).

We need to use Foursquare API to get the venues data within the radius as 10,000m from Berlin city center. There are several parameters which we can pass into the calling URL, so we need to modify to meet our need. As previously analyzed, tourists were those who visited places in Berlin and provide ratings, we will focus on these places. We will pass ***query=tourist*** into the URL to filter these values which will helpful for other tourists.



```
url = 'https://api.foursquare.com/v2/venues/explore?client_id={}&client_secret={}&ll={},{}&v={}&radius={}&limit={}&offset={}
&query=tourist'.format(CLIENT_ID, CLIENT_SECRET, latitude, longitude, VERSION, radius, LIMIT, offset)
```

The dataset we receive contain of the list of interesting places in Berlin



	address	categories	cc	city	country	crossStreet	distance	formattedAddress	id	labeledLatLngs	lat	lng	name	neighborhood	postalCode	state
0	Behrenstr. 55-57	Opera House	DE	Berlin	Deutschland	NaN	188	[Behrenstr. 55-57, 10117 Berlin, Deutschland]	4adcda8af964a520bd4921e3	[{'label': 'display', 'lat': 52.51596828902978...	52.515968	13.386701	Komische Oper	Unter den Linden	10117	Berlin
1	Gendarmenmarkt	Concert Hall	DE	Berlin	Deutschland	Charlottenstr.	429	[Gendarmenmarkt (Charlottenstr.), 10117 Berlin...	4adcda7cf964a520584721e3	[{'label': 'display', 'lat': 52.513652, 'lng':...	52.513652	13.391911	Konzerthaus Berlin	NaN	10117	Berlin
2	Markgrafenstr.	Plaza	DE	Berlin	Deutschland	Mohrenstr.	466	[Markgrafenstr. (Mohrenstr.), 10117 Berlin, De...	4adcda7df964a5206a4721e3	[{'label': 'display', 'lat': 52.51357005399756...	52.513570	13.392720	Gendarmenmarkt	NaN	10117	Berlin
3	Bebelplatz	Plaza	DE	Berlin	Deutschland	Unter den Linden	342	[Bebelplatz (Unter den Linden), 10117 Berlin, ...	4adcda7df964a520ab4721e3	[{'label': 'display', 'lat': 52.51653012045096...	52.516530	13.393847	Bebelplatz	NaN	10117	Berlin
4	Am Festungsgraben 1	Theater	DE	Berlin	Deutschland	NaN	481	[Am Festungsgraben 1, 10117 Berlin, Deutschland]	4adcda89f964a520954921e3	[{'label': 'display', 'lat': 52.51894077519612...	52.518941	13.395245	Maxim Gorki Theater	NaN	10117	Berlin

Figure 1: Foursquare data for Berlin interesting places

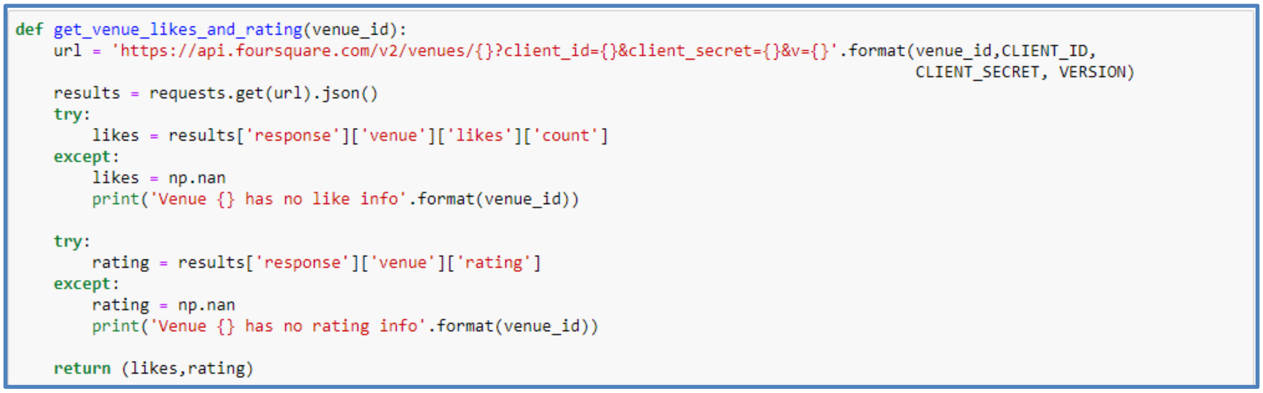
The dataset first includes the following fields:

- address
- categories
- cc (country code)
- city
- country
- crossStreet
- distance
- formattedAddress

- id
- labeledLatLngs
- lat
- lng
- name
- neighborhood
- postalCode
- state

As we use the Foursquare API sandbox account, there is no rating of the place in the returned dataset yet. In order to obtain the ratings, we need to pass the ID of each location to URL, and call the Foursquare API. Since Foursquare provides also the number of likes for each venue, we collect this data in the call, too.

The following function is coded to capture the venue rating and like

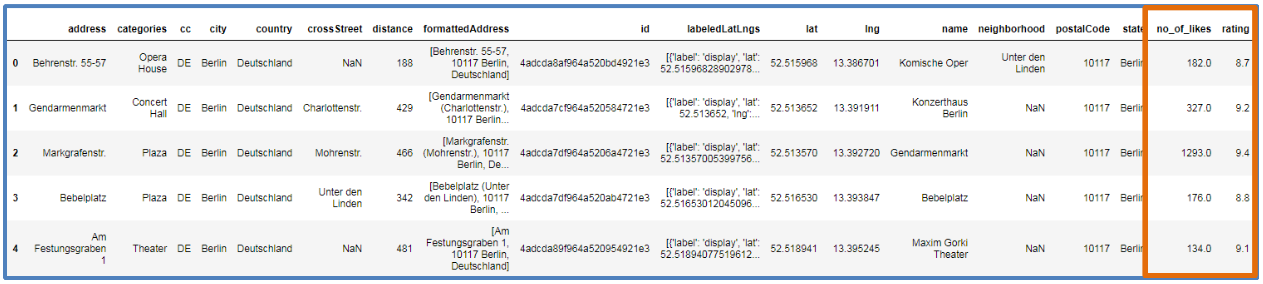


```
def get_venue_likes_and_rating(venue_id):
    url = 'https://api.foursquare.com/v2/venues/{}?client_id={}&client_secret={}&v={}'.format(venue_id,CLIENT_ID,
                                                                                              CLIENT_SECRET, VERSION)

    results = requests.get(url).json()
    try:
        likes = results['response']['venue']['likes']['count']
    except:
        likes = np.nan
        print('Venue {} has no like info'.format(venue_id))

    try:
        rating = results['response']['venue']['rating']
    except:
        rating = np.nan
        print('Venue {} has no rating info'.format(venue_id))

    return (likes,rating)
```

Figure 2: Function to get venue likes and rating from Foursquare

We need to pass each URL for each venue ID and retrieve the information from Foursquare. The results are parsed into JSON file, and we capture the number of likes and ratings from JSON file. The values are then

The consolidated dataset contains the previous dataset, together with the ***rating*** and ***number of likes*** for each venue.



	address	categories	cc	city	country	crossStreet	distance	formattedAddress	id	labeledLatLngs	lat	lng	name	neighborhood	postalCode	state	no_of_likes	rating
0	Behrenstr. 55-57	Opera House	DE	Berlin	Deutschland	NaN	188	[Behrenstr. 55-57, 10117 Berlin, Deutschland]	4adcda8af964a520bd4921e3	[{'label': 'display', 'lat': 52.51596828902978...	52.515968	13.386701	Komische Oper	Unter den Linden	10117	Berlin	182.0	8.7
1	Gendarmenmarkt	Concert Hall	DE	Berlin	Deutschland	Charlottenstr.	429	[Gendarmenmarkt (Charlottenstr.), 10117 Berlin...	4adcda7cf964a520584721e3	[{'label': 'display', 'lat': 52.513652, 'lng':...	52.513652	13.391911	Konzerthaus Berlin	NaN	10117	Berlin	327.0	9.2
2	Markgrafenstr.	Plaza	DE	Berlin	Deutschland	Mohrenstr.	466	[Markgrafenstr. (Mohrenstr.), 10117 Berlin, De...	4adcda7df964a5206a4721e3	[{'label': 'display', 'lat': 52.51357005399756...	52.513570	13.392720	Gendarmenmarkt	NaN	10117	Berlin	1293.0	9.4
3	Bebelplatz	Plaza	DE	Berlin	Deutschland	Unter den Linden	342	[Bebelplatz (Unter den Linden), 10117 Berlin, ...	4adcda7df964a520ab4721e3	[{'label': 'display', 'lat': 52.51653012045096...	52.516530	13.393847	Bebelplatz	NaN	10117	Berlin	176.0	8.8
4	Am Festungsgraben 1	Theater	DE	Berlin	Deutschland	NaN	481	[Am Festungsgraben 1, 10117 Berlin, Deutschland]	4adcda89f964a520954921e3	[{'label': 'display', 'lat': 52.51894077519612...	52.518941	13.395245	Maxim Gorki Theater	NaN	10117	Berlin	134.0	9.1

Figure 3: Dataset with likes and ratings values

III. Methodology

3.1 Exploratory data analysis

The question we are solving is to find the best way to group places into a few major areas, so that in each area we can build a hotel. Since we do not have any target variable, we want to have a way just to group the places naturally; an unsupervised machine learning algorithm will suit better. Clustering is one of the most common exploratory data analysis techniques which often used to get an intuition about the data structure. Clustering can be considered as a task to grouping data points into homogenous subgroups, where data points in the same groups are as similar as possible according to a certain measure, such as Euclidean-based distance. A cluster is a collection of data points aggregated together because of their certain similarities.

3.2 Machine learning algorithm

K-means clustering is the unsupervised machine learning algorithm that we select for this project. k-means clustering is among the most popular and simplest unsupervised machine learning algorithms. Typically, this algorithm help organize the original dataset into k pre-defined distinct clusters.

To implement k-means clustering, we use *sklearn* from python as this library has efficiently taken care of many machine learning algorithms.

IV. Results

4.1. Data exploratory analysis

In this section, we will plot the dataset into appropriate diagram to have the overview about our data. We first plot the number of likes and the ratings of places into histogram

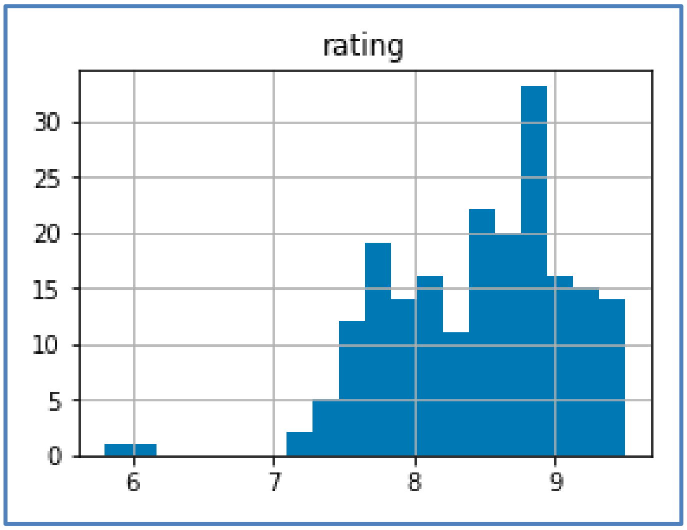


Figure 4: Histogram - rating

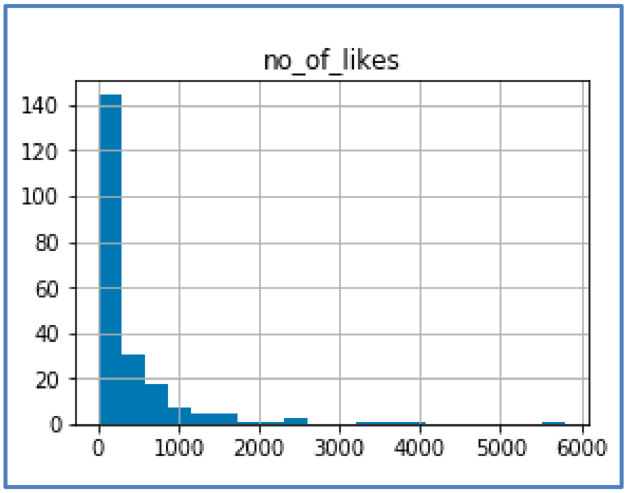
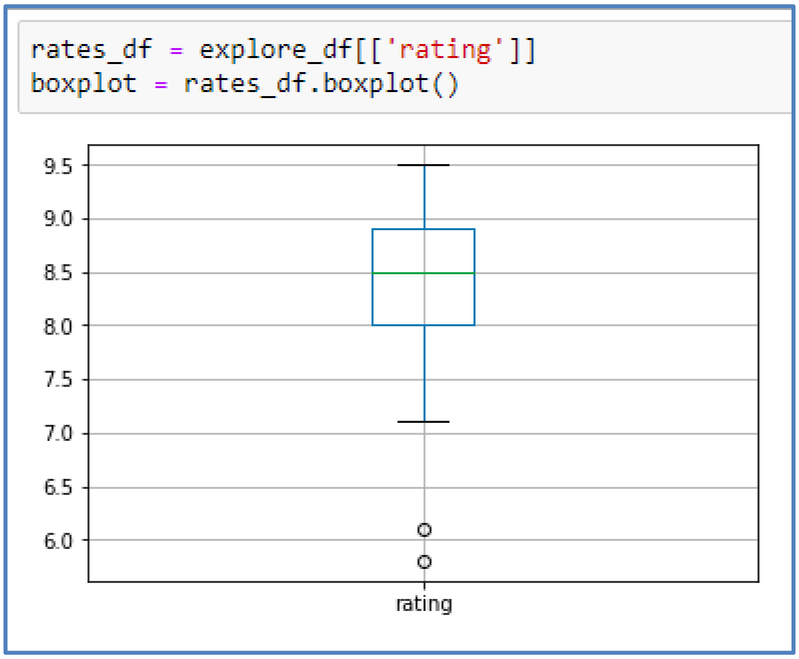


Figure 5: Histogram - number of likes

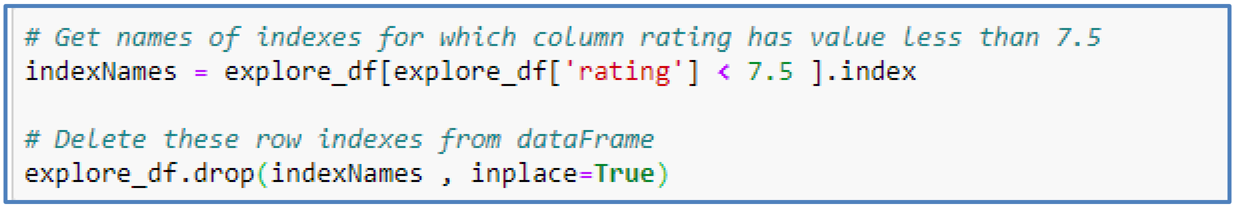
From the histograms, we can see that the majority of places receiving the rating from travelers above 7.0. Most of travelers give the rating around 9.0 for places in Berlin. On the other hands, when exploring the histogram for the distribution of likes, we can observe that it is a left-skewed histogram. Most places will receive around 200-500 likes from people who give feedbacks on Foursquare.

We next explore the boxplot for rating, and from the graph, it is quite obvious that we have some outliers for rating, where the rating is around 8=6.0. The median of ratings is 8.5, while the first quartile value is slightly above 7.0. From this box plot, we can confirm that the assumption we made earlier (Assumption 1) is feasible. We will drop all the places which have rating less than 7.5



```
rates_df = explore_df[['rating']]
boxplot = rates_df.boxplot()
```

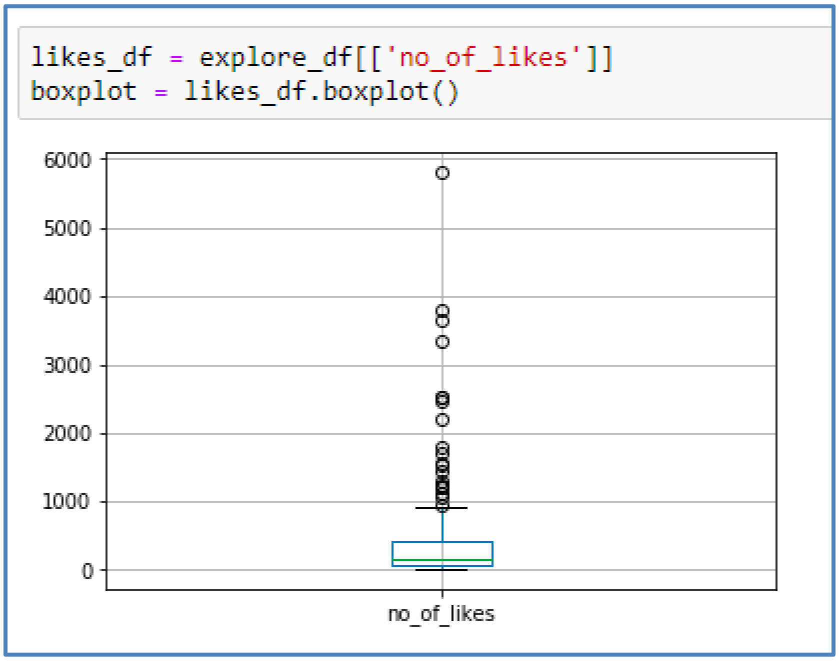
Figure 6: Boxplot - rating



```
# Get names of indexes for which column rating has value less than 7.5
indexNames = explore_df[explore_df['rating'] < 7.5 ].index

# Delete these row indexes from dataFrame
explore_df.drop(indexNames , inplace=True)
```

Figure 7: Code to drop low rating places

When observing the boxplot for number of likes, there are many outliers which represent the places that receive more likes than the average. However, since the outliers on the good site (more likes means the places are more interesting), we remain the data points.



```
likes_df = explore_df[['no_of_likes']]
boxplot = likes_df.boxplot()
```

Figure 8: Box plot - number of likes

4.2. Data visualization

Firstly, we plot all the data points into the map in order to have an overview of our dataset. In the map, the red point represents the City center getting from geocode. The blue points represent all interesting places in Berlin that we retrieved from Foursquare API calls. From the initial visualization, we can see that interesting places can be found almost every corner of the city. We can observe that there is a higher concentration of interesting places near the city center. However, there is no clearly visible cluster in the map.

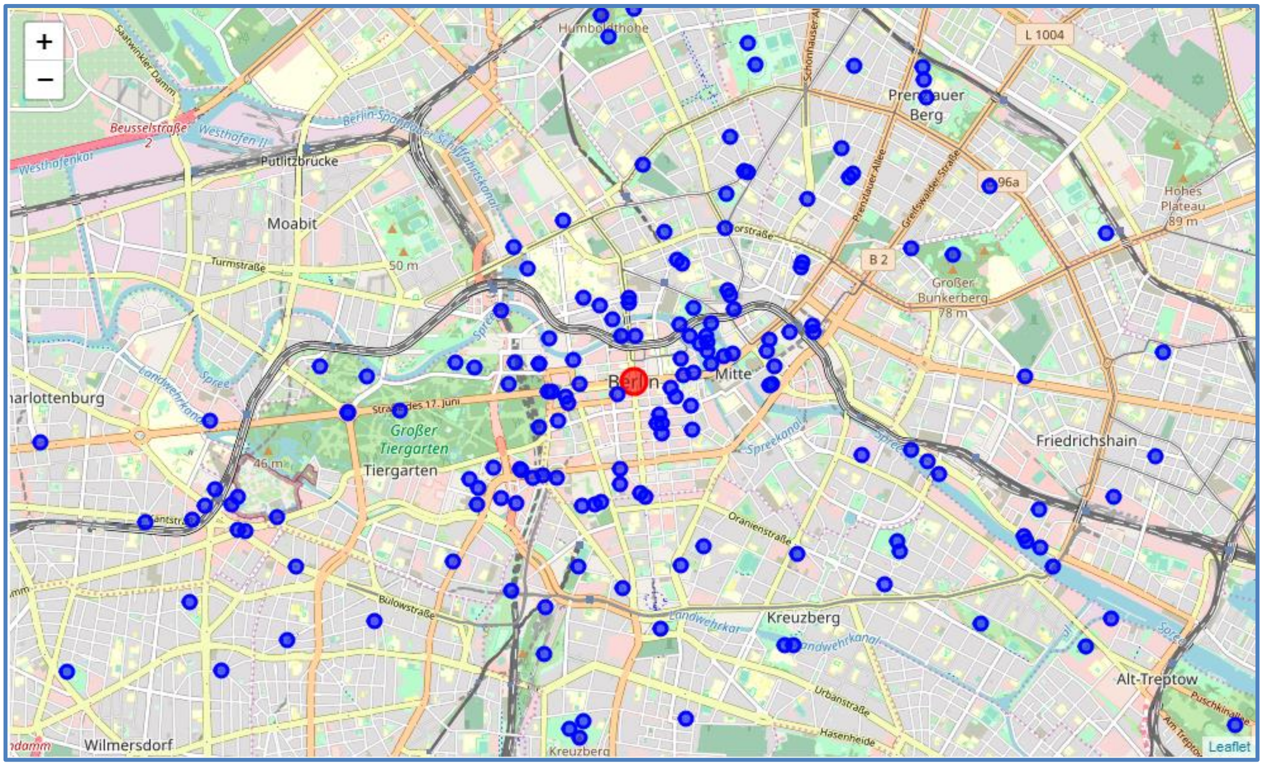


Figure 9: Visualization of dataset

4.3. k-means clustering

After applying k-means clustering with pre-defined k value = 5, the result we achieve from the experiment is the 5 clusters of interesting places in Berlin. We color these 5 clusters by distinct colors as in figure below

Per observation, we could see that red clusters are the one which is the farthest from the city and has the least data points in the cluster. On the other hand, the purple cluster is the one closest to the city center and it seems to contain the most data points in the cluster

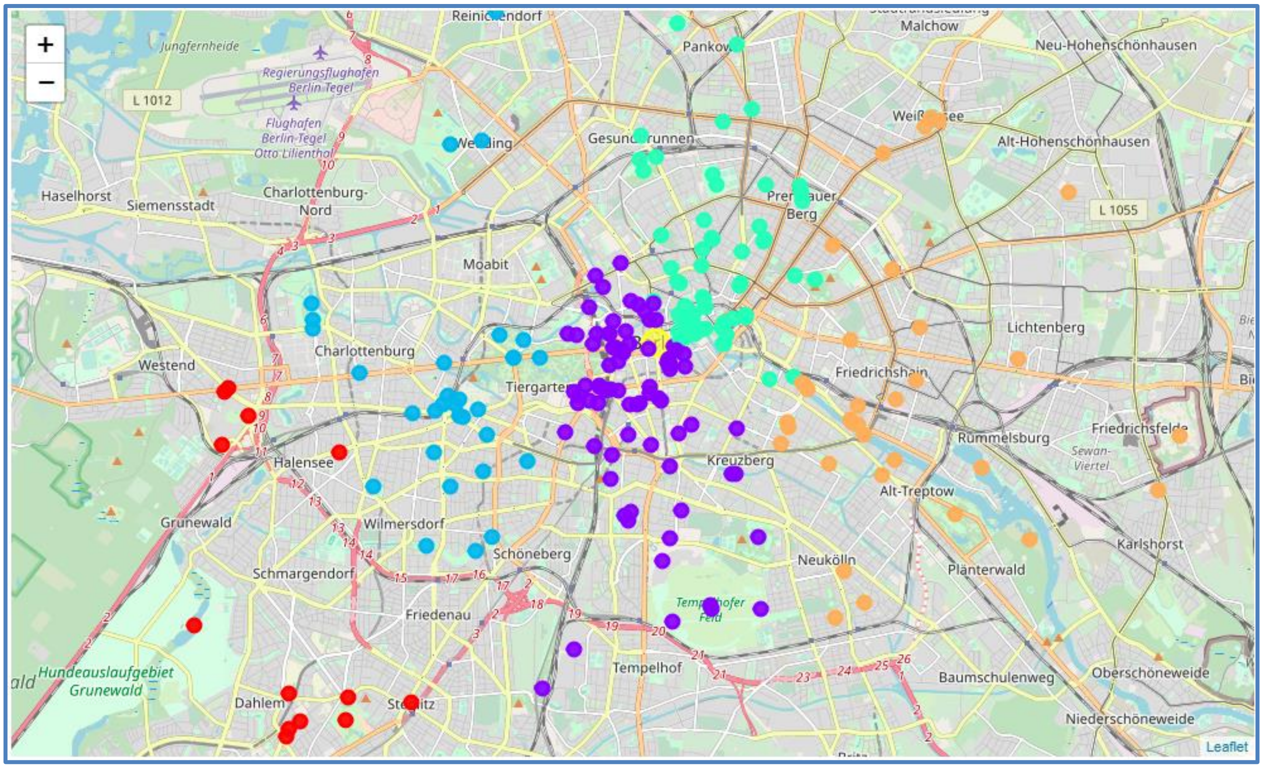


Figure 10: k-means clustering with k = 5

As we plan to build the hotel in centroids of each cluster, we need to use *cluster_centers_* attribute in k-means to obtain the centroids' location data. With the location of centroids, we add the visualization map as shown in the picture below

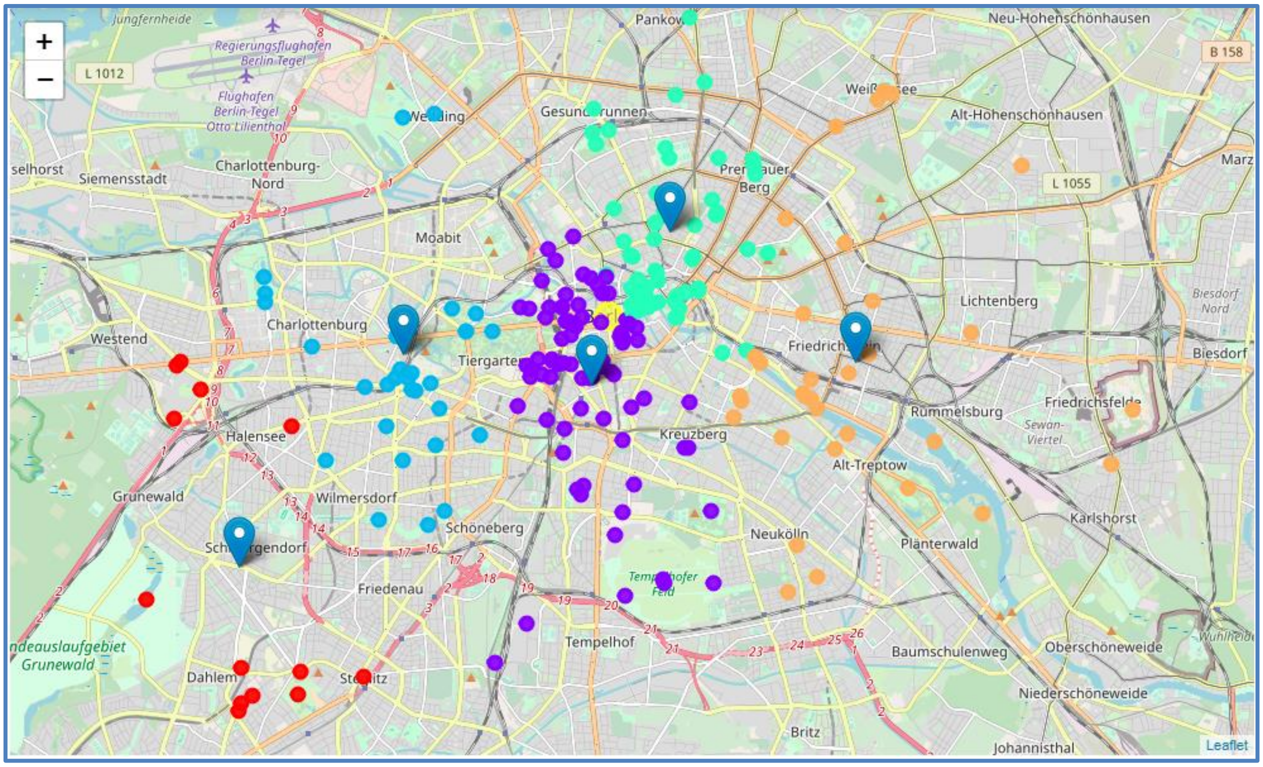


Figure 11: 5 clusters with centroids

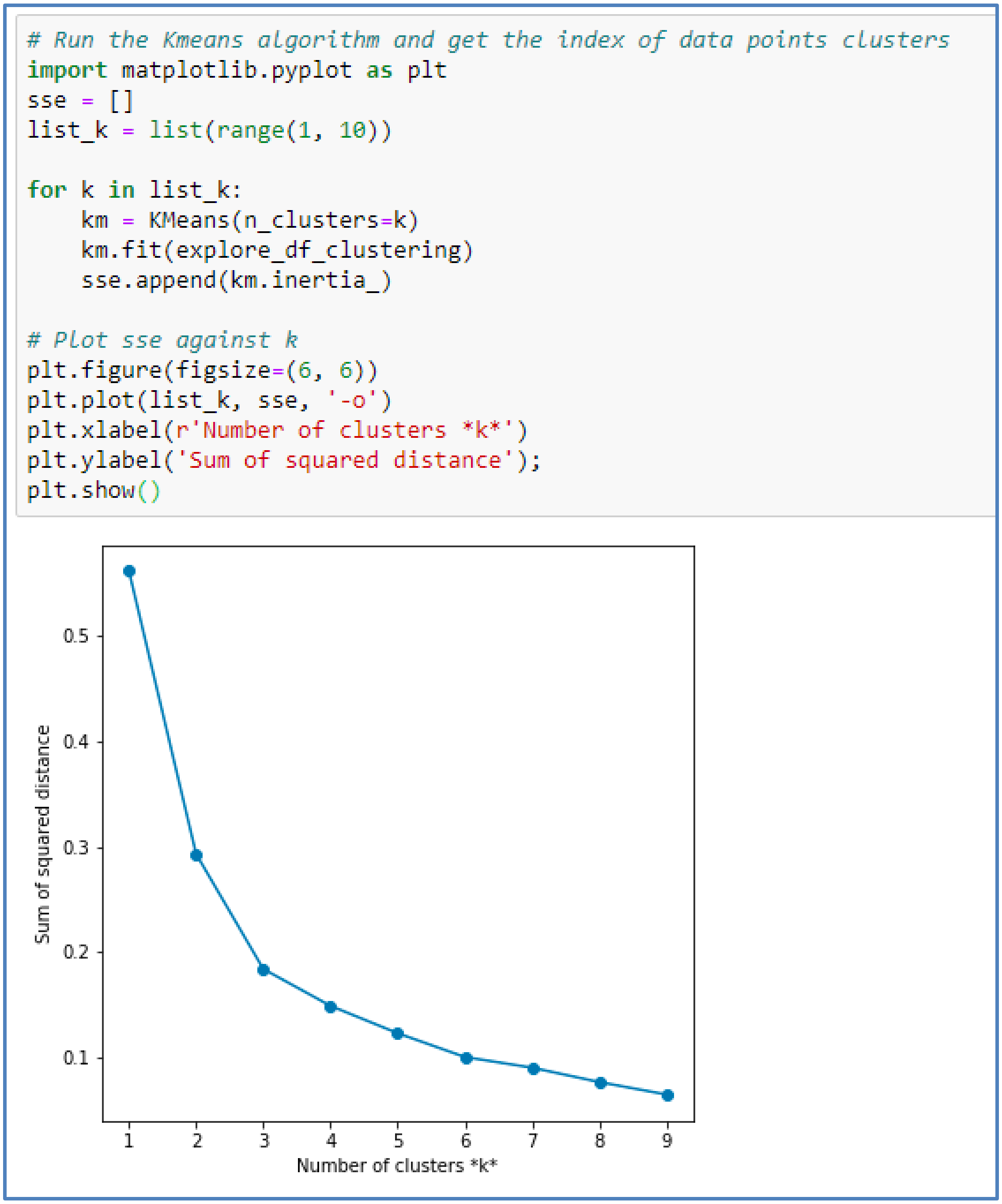
V. Discussion

5.1. Model Evaluation

Elbow method gives us an idea on what a good k number of clusters would be based on the sum of squared distance (SSE) between data points and their assigned clusters' centroids. We pick k at the spot where SSE starts to flatten out and forming an elbow. We'll use the geyser dataset and evaluate SSE for different values of k and see where the curve might form an elbow and flatten out.

Initially, we choose k = 5 with the first assumption that the city can be split into 5 main areas: North, East, South, West and center. For the data evaluation, we will try k clusters from 1-10 to identify which is the k value where the curve starts flattening out.

From the graph displayed, we can see the value k =5 selected earlier indeed is not the best choice for k. It is still pretty hard to figure out the good number of clusters to use because the curve is monotonically decreasing, but we can choose k=3 where we still can see a significant change in the line chart from k =2 to k =3.



```
# Run the Kmeans algorithm and get the index of data points clusters
import matplotlib.pyplot as plt
sse = []
list_k = list(range(1, 10))

for k in list_k:
    km = KMeans(n_clusters=k)
    km.fit(explore_df_clustering)
    sse.append(km.inertia_)

# Plot sse against k
plt.figure(figsize=(6, 6))
plt.plot(list_k, sse, '-o')
plt.xlabel(r'Number of clusters *k*')
plt.ylabel('Sum of squared distance');
plt.show()
```

Figure 12: Model evaluation by Elbow method

5.2. Re-run the model

As we decide to use k =3, we re-run the code to get the newly clusters. From the map, we can see that the interesting places are now grouped into three groups (green, purple and red).

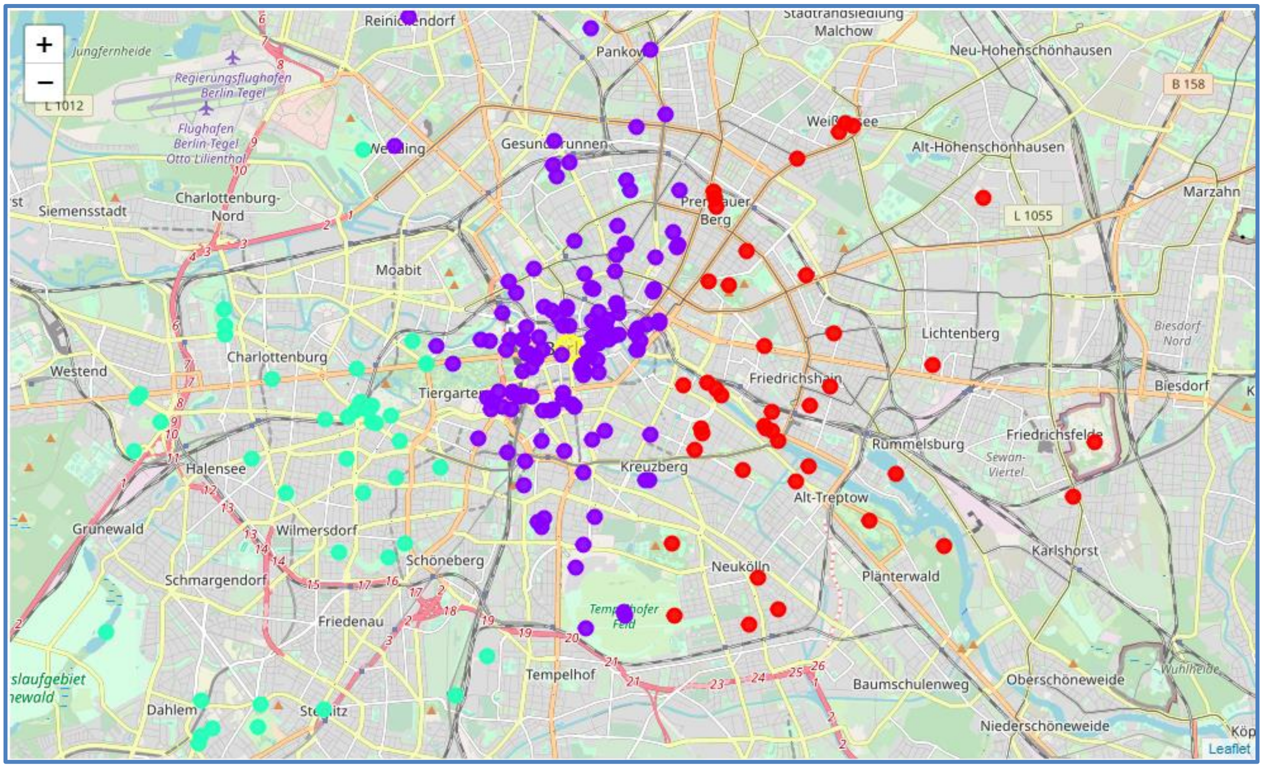


Figure 13: k-means clustering with k = 3

And decide the new centroids for these clusters

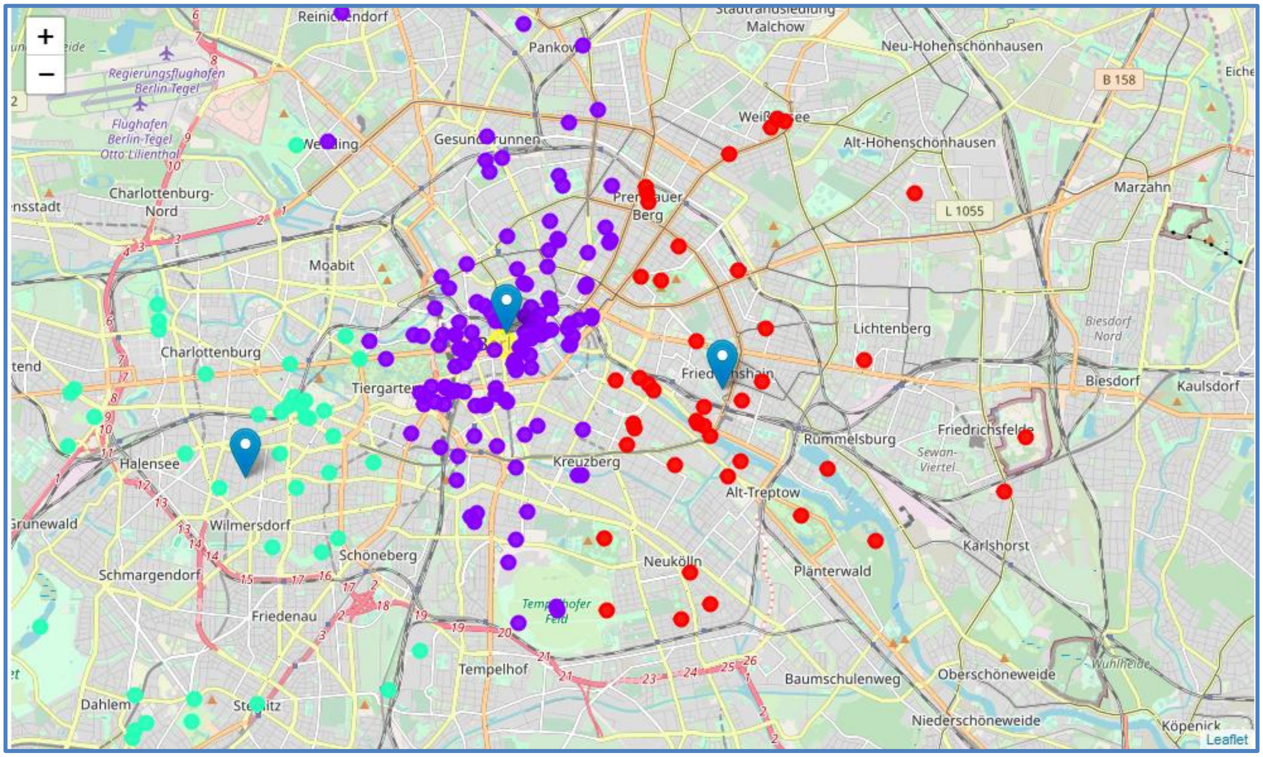


Figure 14: 3 clusters with centroids

From the result, we can recommend the entrepreneur to set up 3 new hotels at the centroids (marked as blue location in the map). If we build the hotels in these places, we can achieve the following goals:

- The positions are in the best locations for all interesting places belonging to the specific clusters, as they locate at the centroids, which were calculated as the mean of longitude and latitude for all data points in that cluster.
- Since we have used the elbow method to evaluate the model, we realize that have 5 clusters does not yield much significant difference with having 3 clusters. As such, instead of building 5 hotels, we can build 3 hotels with approximately same effectiveness.

VI. Conclusion

In this project, we have used an unsupervised machine learning algorithm to tackle the problem. Based on the question, we have identified that k-means clustering is the simplest to implement and can fulfill the requirements for this project. Firstly, we collect data about interesting places in Berlin using Foursquare API, pre-processing it and visualize the data to have the overview about data. Then we apply clustering with k = 5 as the first try. The result is the 5 clusters separated based on the location of interesting places. Moreover, we want to optimize the model by finding the best value of k. We then run the elbow method to find the k value that can make a significant drop in the line graph. We have found out that k = 3 is in fact a suitable number of clusters. So, instead of building 5 hotels, the entrepreneur can build 3 hotels and the capacity to cover interesting places are quite similar. As such, we can see how the business decision can be formed based on the outcome from machine learning.

To sum up, in this project, we have successfully applied an appropriate machine learning algorithm to resolve a business problem. By evaluating the model, we revise to get better model, which also results in the business decision to reduce the number of hotels and invest the capital into more significant works.