PRML MINI PROJECT

IDENTIFYING RESTAURANT HOTSPOTS WITH A GAUSSIAN MIXTURE MODEL

Snigdha Labh, 17070123105
Satyaki Tatte, 17070123112
Ventrapragada Sai Shravani, 17070123120
Vinayak Kuanr, 17070123121

Introduction

Artificial Intelligence and Machine learning have gained a strong foothold across different industries due to their ability to streamline operations, save costs and reduce human error. Machine learning has reshaped sectors like healthcare, finance, and retail, and the restaurant industry is no exception.

When opening a new restaurant, geographical placement is of prime importance in determining whether it will thrive. With the advent of abundant user-generated restaurant reviews, this is potential to leverage these reviews to gain some insights into users' preferences for restaurants.

In our project, we will be identifying restaurant hotspots with Gaussian Mixture Model. We will be plotting and identifying all the restaurants' clusters in Pune based on the location data i.e., latitude and longitude. The idea behind this is to find the clusters that make sense geographically but also have different characteristics in terms of other features like that the location of a restaurant may tell us about the restaurants in certain locations having higher ratings.

Theory

Clustering

Clustering is one of the most common exploratory data analysis technique used to get an intuition about the structure of the data. It can be defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different. In other words, we try to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similarity measure such as Euclidean-based distance or correlation-based distance. The decision of which similarity measure to use is application-specific.

Unlike supervised learning, clustering is considered an unsupervised learning method since we don’t have the ground truth to compare the output of the clustering algorithm to the true labels to evaluate its performance. We only want to try to investigate the structure of the data by grouping the data points into distinct subgroups.

Gaussian mixture models

They are a probabilistic model for representing normally distributed subpopulations within an overall population. Mixture models in general don't require knowing which subpopulation a data point belongs to, allowing the model to learn the subpopulations automatically. Since subpopulation assignment is not known, this constitutes a form of unsupervised learning.

GMMs have been used for feature extraction from speech data, and have also been used extensively in object tracking of multiple objects, where the number of mixture components and their means predict object locations at each frame in sequence.

Gaussian Mixture Model provides better clustering with distinct usage boundaries. Although, Gaussian Mixture Model has higher computation time than K-Means, it can be used when more fine-grained workload characterization and analysis is required.

Geo-clustering

Geo clustering is computer clustering over geographically dispersed sites. A basic cluster is a group of independent computers called nodes, usually housed in the same physical location, that work together to run a common set of applications. The nodes are physically connected by network and storage infrastructure and logically connected by clustering software. Unlike a basic cluster, a geo cluster disperses its nodes to several different physical locations.

A geo cluster is unaware of the physical distance between its nodes.

The **geolocator** will automatically try to request permissions when you try to acquire a location through the getCurrentPosition or getPositionStream methods. We do however provide methods that will allow you to manually handle requesting permissions.

Geo-coding

Geocoding is the process of converting addresses (like a street address) into geographic coordinates (like latitude and longitude), which you can use to place markers on a map, or position the map.

Data Set

Dimension: The data set used for this project is of following dimension 1832rows x 6 columns. This data was downloaded from the Zomato and Kaggle data archive. Though the code maps all 1832 data points (restaurant) in different clusters based on star rating and review count, it takes a long time to execute. So, for demonstration purposes we are only using the first 5 rows of data.

Feature: The longitude and latitude of all the 1832 restaurants were extracted and plotted on a map with the help of Folium & Geopy, libraries available in python.

Python Libraries

Folium: It is a Python library used for visualizing geospatial data. It is easy to use and yet a powerful library. Folium is a Python wrapper for Leaflet.js which is a leading open-source JavaScript library for plotting interactive maps.

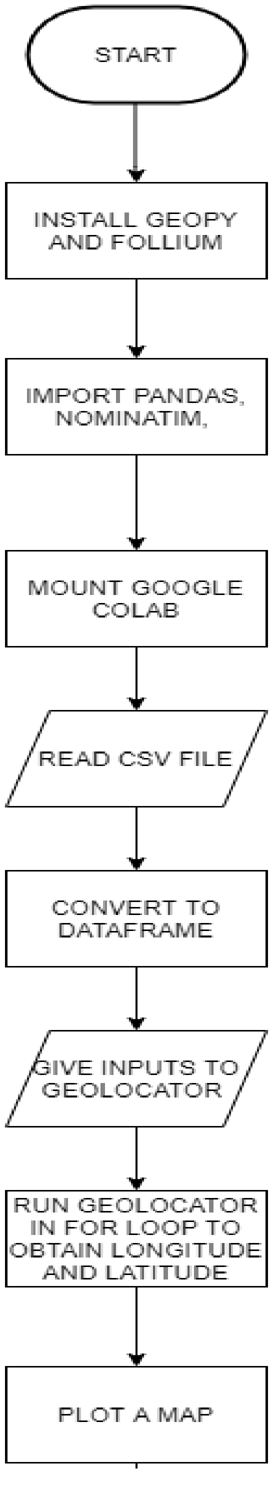
Geopy: It is a Python client for several popular geocoding web services. Geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources.

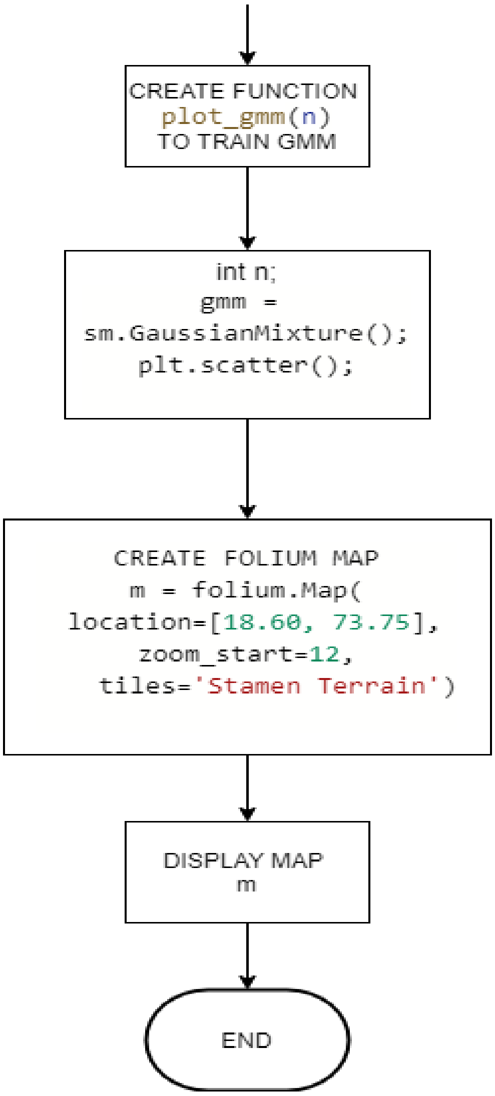
Algorithm

1. Perform step 2 - 6 for Feature extraction
2. Import the necessary libraries
    a. Install geopy and folium
    b. Import pandas, Nominatim and folium
3. Mount google colab and read the csv and convert to dataframe
4. Give inputs to geolocator for getting permissions for geocoding.
5. Run geolocator.geocode in for loop to obtain latitude and longitude of the given address in the dataframe.

6. Plot a map to check if all the given locations are plotted correctly.
7. Create a function to train the GMM and to plot the results
    a. Take a variable n to define the number of clusters
    b. Use the sm.GuassianMixture to fit the gmm
        i. Take the parameters components=n
        ii. Random state=11
        iii. And initial n=5
    c. Plot a scatter plot for the above generated gmm
8. Mark the clusters and the rating using folium map
9. End

Flowchart


START
INSTALL GEOPY AND FOLLIUM
IMPORT PANDAS, NOMINATIM,
MOUNT GOOGLE COLAB
READ CSV FILE
CONVERT TO DATAFRAME
GIVE INPUTS TO GEOLOCATOR
RUN GEOLOCATOR IN FOR LOOP TO OBTAIN LONGITUDE AND LATITUDE
PLOT A MAP



Code

Loading Libraries to be installed

```
!pip install geopy
```

```
Requirement already satisfied: geopy in /usr/local/lib/python3.6/dist-packages (1.17.0)
Requirement already satisfied: geographiclib<2,>=1.49 in /usr/local/lib/python3.6/dist-packages
(from geopy) (1.50)
```

```
!pip install folium
```

```
Requirement already satisfied: folium in /usr/local/lib/python3.6/dist-packages (0.8.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.6/dist-packages (from folium)
(2.11.2)
Requirement already satisfied: requests in /usr/local/lib/python3.6/dist-packages (from folium)
(2.23.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from folium)
(1.18.5)
```

```
Requirement already satisfied: branca>=0.3.0 in /usr/local/lib/python3.6/dist-packages (from
folium) (0.4.1)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages (from folium) (1.15.0)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.6/dist-packages (from
jinja2->folium) (1.1.1)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.6/dist-packages (from
requests->folium) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
/usr/local/lib/python3.6/dist-packages (from requests->folium) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.6/dist-packages (from
requests->folium) (2020.6.20)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.6/dist-packages (from
requests->folium) (3.0.4)
```

```
import pandas as pd
from geopy.geocoders import Nominatim
import folium
```

```
# loading the dataset from google drive
from google.colab import drive
drive.mount('/content/drive')
path = "/content/drive/My Drive/addressbook_for_lat_long.csv"
df = pd.read_csv("/content/drive/My Drive/addressbook_for_lat_long.csv")
df
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

Sr.No.	Locality	City	LocalityCity	Ratings_out_of_5	Number of votes	
0	0	Hinjawadi	, Pune, India	Hinjawadi, Pune, India	4.9	7029 votes
1	1	Hinjawadi	, Pune, India	Hinjawadi, Pune, India	4.3	430 votes
2	2	Wakad	, Pune, India	Wakad, Pune, India	4.6	4731 votes
3	3	Wakad	, Pune, India	Wakad, Pune, India	4.5	2999 votes

4	4	Hinjawadi	, Pune, India	Hinjawadi, Pune, India	4	1437 votes
...	...	...	...	...	...	..
1827	1827	Wakad	, Pune, India	Wakad, Pune, India	0	
1828	1828	Aundh	, Pune, India	Aundh, Pune, India	0	
1829	1829	Wakad	, Pune, India	Wakad, Pune, India	0	
1830	1830	Wakad	, Pune, India	Wakad, Pune, India	0	
1831	1831	Hinjawadi	, Pune, India	Hinjawadi, Pune, India	0	

1832 rows × 6 columns

```
geolocator = Nominatim(timeout=10, user_agent = "Your_Name")
```

Feature Extraction

Geocode the addresses in the Pandas Dataframe

The following occurs in this cell:

- **Iterates over all rows in the dataframe df.**
- **Submits the string we just created as the address to be geocoded**
- **Saves results to the geocodes list object.**

```
geocodes = [geolocator.geocode(df['LocalityCity'][x])for x in range(len(df))]
geocodes
```

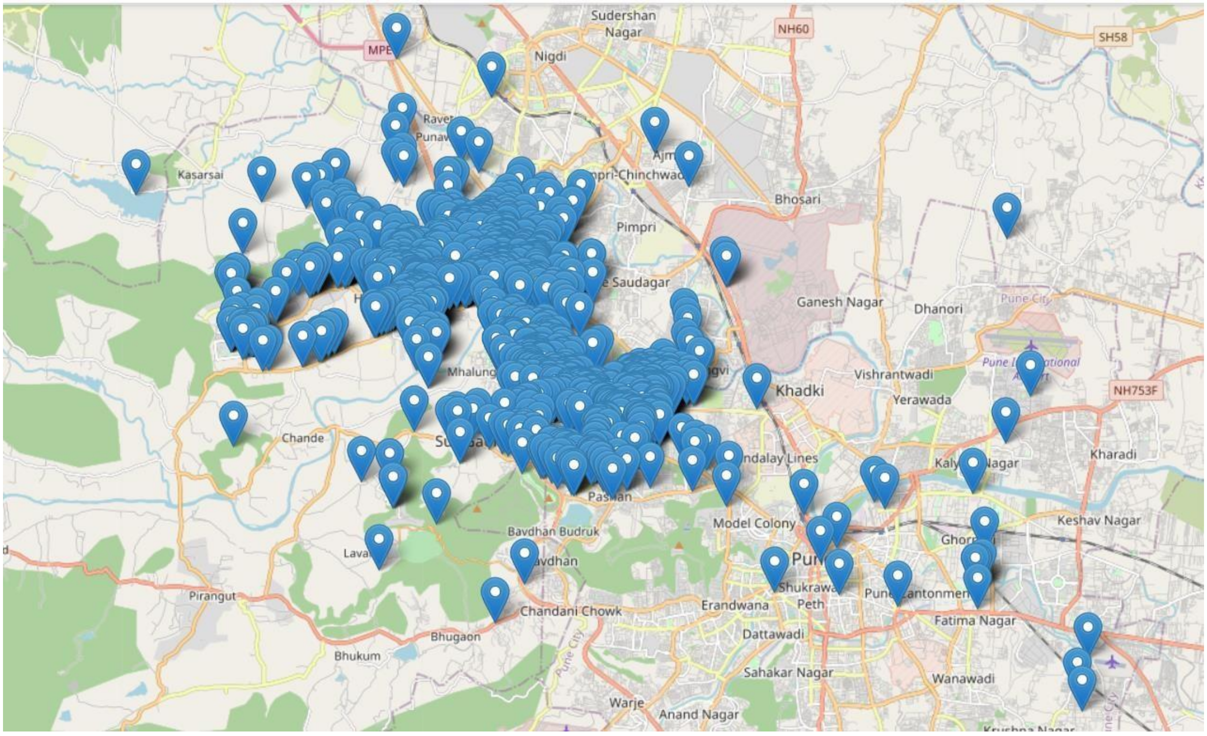
```
df['lat'] = [g.latitude for g in geocodes]
df['lon'] = [g.longitude for g in geocodes]
df.dropna(axis=0)
df
```

Sr.No.	Locality	City	LocalityCity	Ratings_out_of_5	Number of votes	lat	lon	
0	0	Hinjawadi	, Pune, India	Hinjawadi, Pune, India	4.9	7029 votes	18.594063	73.742049
1	1	Hinjawadi	, Pune, India	Hinjawadi, Pune, India	4.3	430 votes	18.594063	73.742049
2	2	Wakad	, Pune, India	Wakad, Pune, India	4.6	4731 votes	18.610902	73.763796
3	3	Wakad	, Pune, India	Wakad, Pune, India	4.5	2999 votes	18.610902	73.763796
4	4	Hinjawadi	, Pune, India	Hinjawadi, Pune, India	4	1437 votes	18.594063	73.742049
...	...	...	...	...	...	...	...	...
1827	1827	Wakad	, Pune, India	Wakad, Pune, India	0	0	18.610902	73.763796
1828	1828	Aundh	, Pune, India	Aundh, Pune, India	0	0	18.561883	73.810196
1829	1829	Wakad	, Pune, India	Wakad, Pune, India	0	0	18.610902	73.763796
1830	1830	Wakad	, Pune, India	Wakad, Pune, India	0	0	18.610902	73.763796
1831	1831	Hinjawadi	, Pune, India	Hinjawadi, Pune, India	0	0	18.594063	73.742049

1832 rows × 8 columns

```
map1 = folium.Map(location=(18.9, 73.3), zoom_start=12)
```

```
for index,row in df.iterrows():
  # Add the geocoded locations to the map
  folium.Marker(location=(row['lat'],row['lon']), popup=row['LocalityCity']).add_to(map1)

display(map1)
```



Importing Libraries

```
import sklearn.mixture as sm
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline
```

Importing dataset

```
from google.colab import drive
drive.mount('/content/drive')
path = "/content/drive/My Drive/dataset2.csv"
data = pd.read_csv("/content/drive/My Drive/dataset2.csv")
data
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

	Restaurant_Name	Detail_address	Latitude	Longitude	review_count	stars

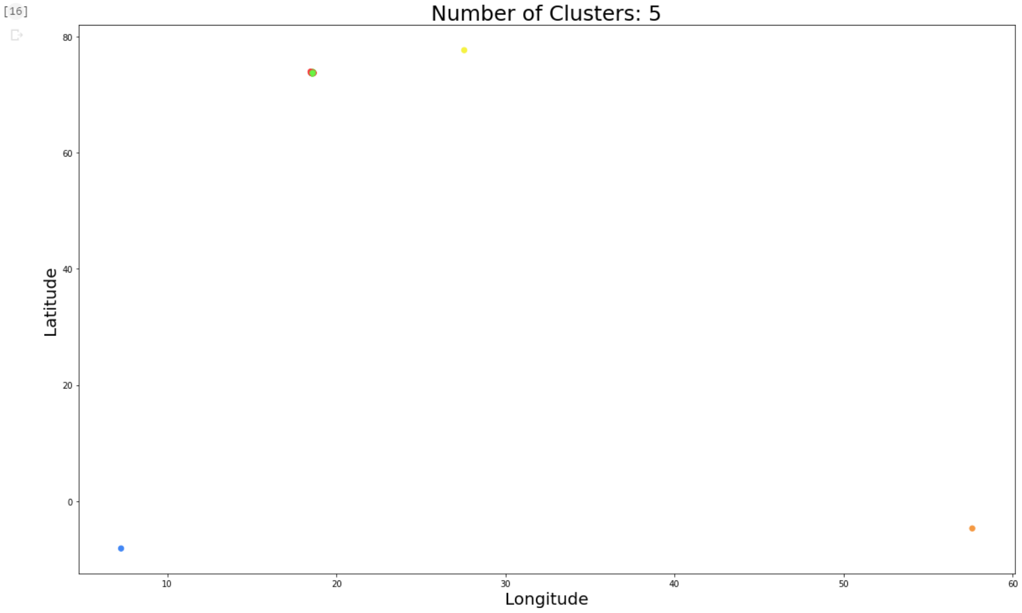
0	AB's - Absolute Barbecues	White Square Building, Hinjawadi, Pune	18.590904	73.753085	7029	5
1	I Amsterdam	Survey 257/1/1A, Near Raj Laxmi Petrol Pump, P...	18.594911	73.728838	430	4
2	Flechazo	165, 3rd Floor, Vantagio, Near Silver Sports C...	18.592662	73.759950	4731	5
3	Barbeque Nation - Sayaji Hotel	Sayaji Hotel, 10th Floor, 135/136, Mumbai Bang...	18.599722	73.754868	2999	5
4	BeHive	New DP Road, Near Laxmi Chowk, Hinjawadi, Pune	18.606650	73.730624	1437	4
...	...	...	...	...	...	...
1827	Superman's Cafe	Shop 3, Near Indira College, Tathwade, Wakad, ...	18.609633	73.745996	0	0
1828	The Chinese Box	Flat 3, 1st Floor, Balaji Niwas, Sharda Park, ...	18.560147	73.802701	0	0
1829	Not Just Parathas	Hinjewadi Road, Kaspate Corner, Wakad, Pune	18.588410	73.768740	0	0

1830	Southentic	Wakad Kaspate Corner, Wakad, Pune	18.590364	73.768606	0	0
1831	Nil Food Point	Mukai Nagar, Hinjawadi, Pune	18.596007	73.731653	0	0

1832 rows × 6 columns

Fitting the GMM

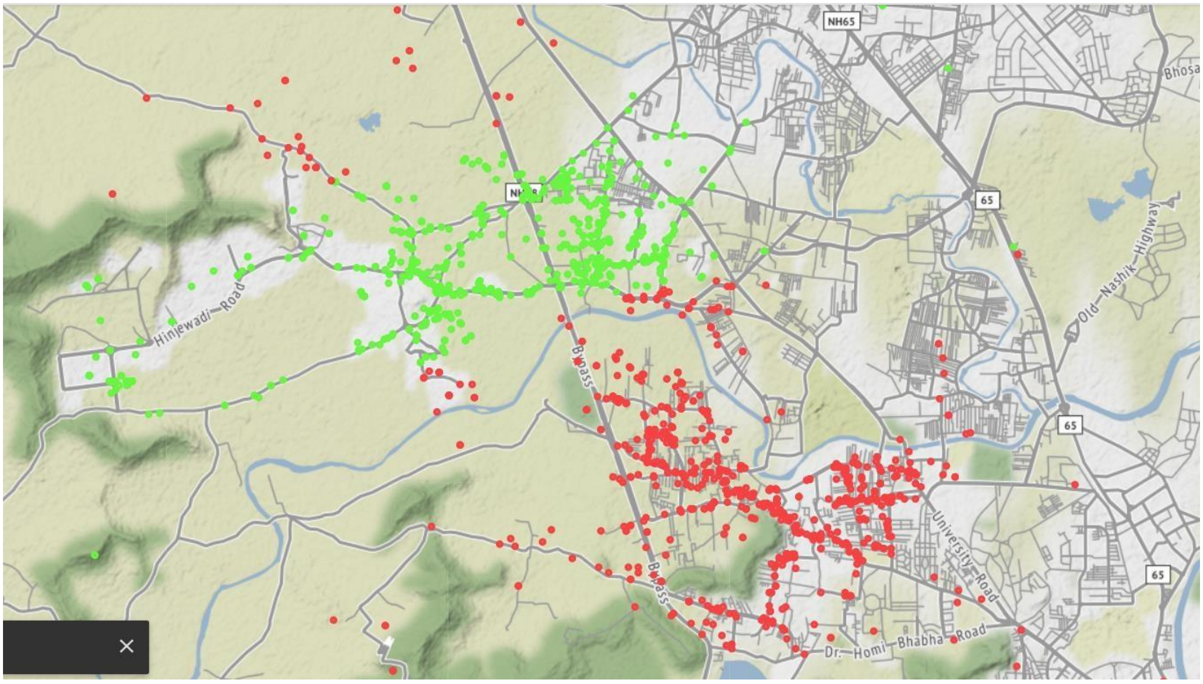
```
def plot_gmm(n):
    # Train a GMM and plot the results

    # Parameters
    n : int
        # Number of clusters

    #Fit gmm and get labels
    x = data[['lat','lon']].values
    gmm = sm.GaussianMixture(n_components=n, random_state=11,n_init=5)
    labels = gmm.fit(x).predict(x)

    #assign a colour to each label
    colour = ['#f54242','#4287f5','#f59942','#f5f242','#69f542','#b342f5']
    c = [colour[l] for l in labels]

    #scatter plot
    plt.figure(figsize=(20, 12))
    plt.scatter(x=x[:, 0], y=x[:, 1], c=c, s=40, cmap='Set1', zorder=1)
    plt.title('Number of Clusters: {}'.format(n),size=25)
    plt.xlabel('Longitude',size=20)
    plt.ylabel('Latitude',size=20)

plot_gmm(5)
```



Analysis of clusters

```
#Get gmm predictions
x = data[['longitude','latitude']].values
gmm = sm.GaussianMixture(n_components=5, random_state=11,n_init=5)
labels = gmm.fit(x).predict(x)

#create folium map
m = folium.Map(
    location=[43.77923, -79.41731999999998],
    zoom_start=12,
    tiles='Stamen Terrain'
)

colour = ['#f54242','#4287f5','#f59942','#f5f242','#69f542','#b342f5']

#add markers to map
for i in range(len(x)):
    lon = x[i][0]
    lat = x[i][1]
    label = labels[i]

    #assign colour based on label
    c = colour[label]

    #add marker
    folium.CircleMarker(location=[lat,lon],
                        radius=2,
                        color=c,
                        fill_color=c).add_to(m)
```

```
#display map
m
```



Summary

In the above programme, we have created a code using gmm, it clusters the restaurants in Pune and shows them on the map with their locations.

Conclusion

In conclusion, the most important advantage of GMM is that it is more flexible in terms of cluster variance and covariance. Firstly, more variance flexibility means that GMMs can better identify clusters with unequal variances. In other words, it will give better results when we can have both clusters that are dense and clusters that are spread out.

Geocoding is a critical task in many location tasks that require coordinate systems. Google Maps geocoding services are more powerful than conventional services like FourSquare, etc.

Thus in this project, the clustering algorithm successfully divides the map into 7 neighborhoods and creates a heat map of around 1800 restaurants based on aggregator reviews and star ratings mainly functioning on latitude and longitude of the locations.