
PROBLEM: USING MACHINE LEARNING TO SUGGEST FOOD, COFFEE, NIGHTLIFE, FUN AND SHOPPING PLACES TO TOURISTS

Abhishek Paul

National Institute of Technology, Durgapur
Github Profile: <https://github.com/abhishekpawl>

1 Introduction

Business Problem:

The objective of this project is to suggest tourists visiting Kolkata, India places for food, coffee, nightlife, fun and shopping based on the location of the hotel in which they will be staying to save their time making plans. I have used an *Unsupervised Machine Learning* method, namely, **Clustering** for recommending the tourists places. This project aims to answer the question: What are the places for food, coffee, nightlife, fun and shopping that tourists can try out?

Background:

For this capstone project, I am concentrating on the scenario of tourists visiting Kolkata, India. Kolkata is called 'the city of joy'. The city is known for its grand colonial architecture, art galleries, cultural festivals, sweets, food culture and many more. Kolkata is a common destination

for vacations. This city attracts tourists from all over the world. So, this project is concerned about suggesting tourists places where they can hangout such as places for food, coffee, nightlife, fun and shopping and thereby saving their valuable vacation-time. This project takes into account the geocoordinates (latitude, longitude) of the hotel and based on the same, makes the suggestions.

Target Audience:

The target audience for this project are the tourists visiting spending their vacation in Kolkata, IN.

Clustering:

Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. The particular method used in the project is k-means clustering. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori. The main idea is to define k centers, one for each cluster. The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed. At this point we need to re-calculate k new centroids as barycentre of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new center. A loop has been generated. As a result of this loop we may notice that the k centers change their location step by step until no more changes are done or in other words centers do not move any more.

2 Data

All the data used in the project are fetched using the **FOURSQUARE API**.

Data for hotels:

```
'https://api.foursquare.com/v2/venues/search?client_id={ }&client_secret={ }&ll={ },{ }&v={ }&query={ }&radius={ }&limit={ }'.format(CLIENT_ID, CLIENT_SECRET, Latitude, Longitude, VERSION, 'Hotel', RADIUS, LIMIT)
```

Data for food places near the hotel:

```
'https://api.foursquare.com/v2/venues/search?client_id={ }&client_secret={ }&ll={ },{ }&v={ }&query={ }&radius={ }&limit={ }'.format(CLIENT_ID, CLIENT_SECRET, HOTEL_Latitude, HOTEL_Longitude, VERSION, 'Food', RADIUS, LIMIT)
```

Data for coffee places near the hotel:

```
'https://api.foursquare.com/v2/venues/search?client_id={ }&client_secret={ }&ll={ },{ }&v={ }&query={ }&radius={ }&limit={ }'.format(CLIENT_ID, CLIENT_SECRET, HOTEL_Latitude, HOTEL_Longitude, VERSION, 'Coffee', RADIUS, LIMIT)
```

Data for nightlife places near the hotel:

```
'https://api.foursquare.com/v2/venues/search?client_id={ }&client_secret={ }&ll={ },{ }&v={ }&query={ }&radius={ }&limit={ }'.format(CLIENT_ID, CLIENT_SECRET, HOTEL_Latitude, HOTEL_Longitude, VERSION, 'Nightlife', RADIUS, LIMIT)
```

Data for fun places near the hotel:

```
'https://api.foursquare.com/v2/venues/search?client_id={ }&client_secret={ }&ll={ },{ }&v={ }&query={ }&radius={ }&limit={ }'.format(CLIENT_ID, CLIENT_SECRET, HOTEL_Latitude, HOTEL_Longitude, VERSION, 'Fun', RADIUS, LIMIT)
```

Data for shopping places near the hotel:

```
'https://api.foursquare.com/v2/venues/search?client_id={ }&client_secret={ }&ll={ },{ }&v={ }&query={ }&radius={ }&limit={ }'.format(CLIENT_ID, CLIENT_SECRET, HOTEL_Latitude, HOTEL_Longitude, VERSION, 'Shopping', RADIUS, LIMIT)
```