## **Battle of the Neighbourhoods**

Aritra Koley

28th March, 2019

1. **Introduction**
   1. **Background**

A major part of the Indian population has to move to a different city to pursue a career. Most people who need to settle outside their own cities and towns are college freshers with little to no experience of living in an unfamiliar environment. Leaving behind one’s hometown to settle in a different city is a daunting task to most people and finding a similar neighbourhood to settle in, which provides the basic facilities that their hometown had just makes things easier.

* 1. **Problem**

The aim of this workflow is to find areas that are similar in terms of facilities and venues but are not in the same city. Two areas will be deemed similar if the areas have similar categories of venues within a similar distance. Locations with such similarity will be grouped into the same cluster. The primary goal after forming such clusters is to visualize them as same distinctly coloured points with the colour representing the cluster they belong to.

* 1. **Interest**

The people who will be benefited are those trying to move to a new city but would like the comforts of their old neighbourhood to make adjusting to a new city easier. Knowing which locations are similar in terms of venues will also be helpful for those trying to expand their business into newer zones as they can use data from where their business made more profits to decide where to expand to.

1. **Data Acquisition and Cleaning**
   1. **Data Sources**

The types of data required could not be collected from a single source so different parts of it were collected from different sources. The two locations to be compared, i.e. the names of the cities, the states they belong to and the country they are part of were all taken as hardcoded parameters for simplicity. In a realistic scenario, they will be considered ‘user input’. The latitudes and longitudes of the chosen cities were obtained by querying a suitable geocoder service, for example, Geolake or Nominatim. The pincodes for each city were obtained by scraping the [www.mapsofindia.com](http://www.mapsofindia.com) website and filtering out the required data for the given cities from html tables. For example,

|  |  |  |  |
| --- | --- | --- | --- |
| Location | Pincode | State | District |
| AI staff colony | 400029 | Maharashtra | Mumbai |
| Aareymilk Colony | 400065 | Maharashtra | Mumbai |
| … | … | … | … |

The latitudes and longitudes for each pincode in each city was obtained by querying a geocoding service, respecting their rate limit regulations and terms of service. Geolake seemed to provide the required data with relative ease but any other service could have been used in its place.

The data for venues in and around a given location was obtained by using the Foursquare API to query for top venues. The foursquare data is retrieved as JSON strings from API calls.

For example,

{

"name": "Harry's Italian Pizza Bar",

"location": {

"address": "225 Murray St",

"lat": 40.71521779064671,

"lng": -74.01473940209351,

"labeledLatLngs": [{

"label": "display",

"lat": 40.71521779064671,

"lng": -74.01473940209351

}],

"distance": 58,

"postalCode": "10282"

}

}

* 1. **Data Cleaning**

Most of the data obtained from the above-mentioned sources needed some sort of preprocessing before they could be fed into a machine learning or data analysis algorithm. If the names of the cities were received from the user in the form of strings, they would need to be formatted into a list of dictionaries format for ease of access to various details about the chosen cities.

For the, pincode tables scraped from the mapsofindia website, only the required table was converted into a pandas dataframe and unnecessary columns like ‘Location’, ‘District’ and ‘State’ were dropped. New columns to store the retrieved latitudes and longitudes were added.

The Foursquare JSON data comes with a lot of information that we do not require hence only the required keys were retrieved from the JSON string. This retrieved JSON was then flattened into a pandas dataframe and all information not pertaining to the venues and their categories were removed.

* 1. **Feature Selection**

Venue names are varied but what determines the characteristics of a location are the categories for the venues in that region. Therefore, the best feature to determine the type of a location are the categories of its top venues and the number of venues of the same category that occur within a given radius of that region. Therefore, the categories found in both cities were one-hot encoded to generate a feature-set for further application of a machine learning algorithms. The mean of the one-hot encoded tuples grouped by the pincodes were taken as the feature set for clustering and classification.

1. **Modelling**

The approach taken to modelling the problem using machine learning algorithms once the feature sets were obtained can be summarized in a simple two-step process.

* 1. **Clustering Points in Base City**

The first phase uses simple K-Means clustering to group similar locations in the base city. As mentioned above, the clustering was done based on the features obtained in the previous feature selection step. Only the data corresponding to the Base City was used to perform clustering so that the various classes of the locations can be determined before similarity analysis with the target city. The value of K was set to 5 for the demonstrative purposes of this report. Therefore, in this phase the classes were determined.

* 1. **Classifying points in Target City**

The second phase uses the information gained form the first phase to determine which locations in the target city are similar to the corresponding locations in the base city. The classification technique used in the second phase is simple K-Nearest Neighbour approach with a value of K set to 5 and weights taken as the inverse of the distance between the points. Therefore, in this case the locations in the target city were classified based on the determined classes in the clustering phase.

1. **Results**

The following images show the two cities with similar locations indicated with identically coloured points. The cities used in the example are Chennai and Kolkata with are both in India.

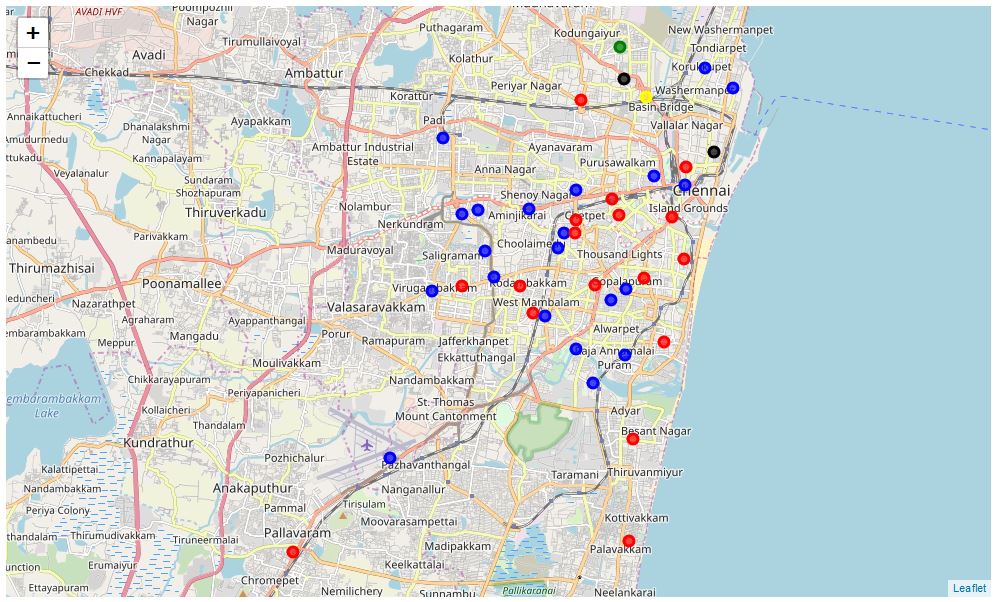


Figure 1: Clustered locations in Chennai

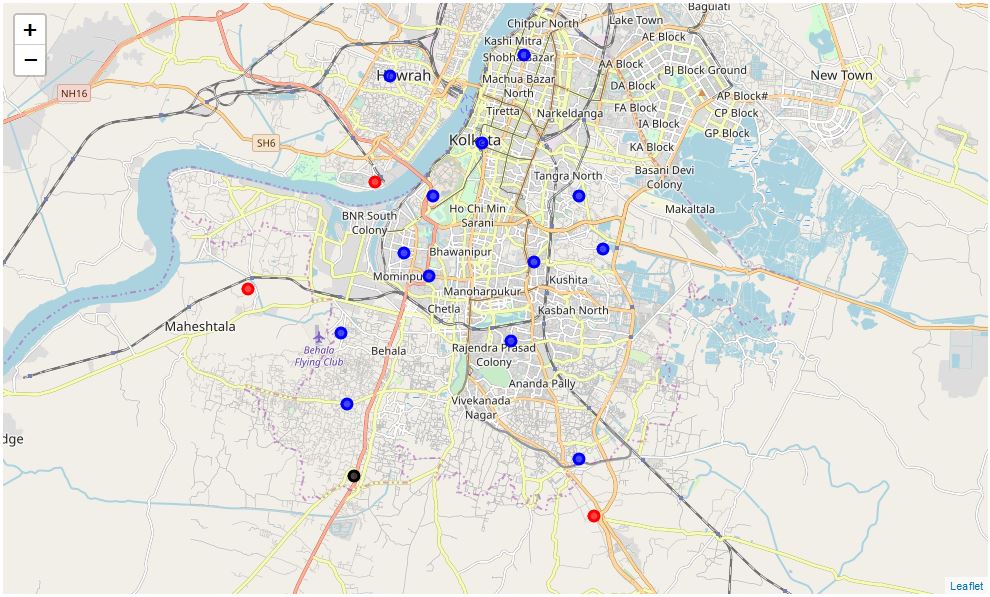


Figure 2: Classified Locations in Kolkata based on Locations in Chennai

1. **Conclusions**

The results show that the workflow is functional and is capable of detecting similar locations in two different cities. From the above we see that there are certain locations in Chennai that do not have suitable counterparts in Kolkata. The class coloured blue is the most common type of location in both the cities with red coming a close second. The black coloured locations are uncommon but present in both cities but towards the outskirts which might indicate either suburban or industrial associations. More in-depth analysis will be required to determine the properties of each cluster as distinct form the other ones but for a preliminary similarity assessment of the locations of two cities, that sort of in-depth analysis is out of the scope of this report.

1. **Future Directions**

Further work can be done to enhance the feature selection based on other factors apart from just popular venues. Determination of the properties of each cluster and fine-tuning of the parameters for the algorithms or even using new innovative algorithms to determine classes and then classify based on them are possible in later stages.

It might be possible to develop this kind of a service into an app that allows access to information and analysis of this sort readily instead of a machine learning workflow format.