



Crio.Do

## Exploratory Analysis of Geolocational Data

### Project Description

Experience a day in the life of a Data Scientist/Engineer by doing a full-fledged analysis of real-life data. Fetch, Clean, Analyse and run K Means Clustering on Geolocational data to recommend accommodations to immigrants to a city!

### Author

[Anant Shukla](#)

### Collaborator(s)

[Kevin Paulose](#), [Kiran Suresh](#)

### Project Language(s)

Python

### Difficulty

Intermediate

### Duration

15 hours

### Prerequisite(s)

Python

### Skills to be learned

REST API, K-Means Clustering, Data Preparation, Data Visualisation

### Overview

### Objective

This project involves the use of K-Means Clustering to find the best accommodation for students in Bangalore (or any other city of your choice) by classifying accommodation for incoming students on the basis of their preferences on amenities, budget and proximity to the location.

## Project Context

Implementing the project will take you through the daily life of a data science engineer - from data preparation on real-life datasets, to visualising the data and running machine learning algorithms, to presenting the results.

In the fast-moving, effort-intense environment that the average person inhabits, It's a frequent occurrence that one is too tired to fix oneself a home-cooked meal. And of course, even if one gets home-cooked meals every day, it is not unusual to want to go out for a good meal every once in a while for social/recreational purposes. Either way, it's a commonly understood idea that regardless of where one lives, the food one eats is an important aspect of the lifestyle one leads.

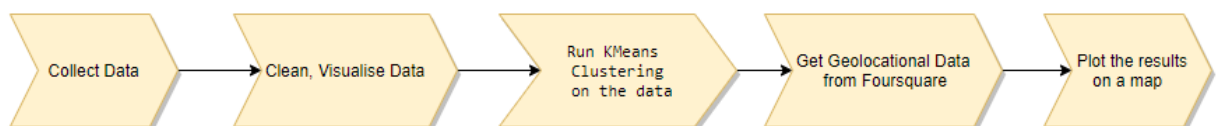
Now, imagine a scenario where a person has newly moved into a new location. They already have certain preferences, certain tastes. It would save both the student and the food providers a lot of hassle if the student lived close to their preferred outlets. Convenience means better sales, and saved time for the customer.

Food delivery apps aside, managers of restaurant chains and hotels can also leverage this information. For example, if a manager of a restaurant already knows the demographic of his current customers, they'd ideally want to open at a location where this demographic is at its highest concentration, ensuring short commute times to the location and more customers served. If potential hotel locations are being evaluated, a site that caters to a wide variety of tastes would be ideal, since one would want every guest to have something to their liking.

This project is a good start for beginners and a refresher for professionals who have dabbled in python / ML before. The methodology can be applied to any location of one's choosing, so feel free to innovate!

## Project Stages

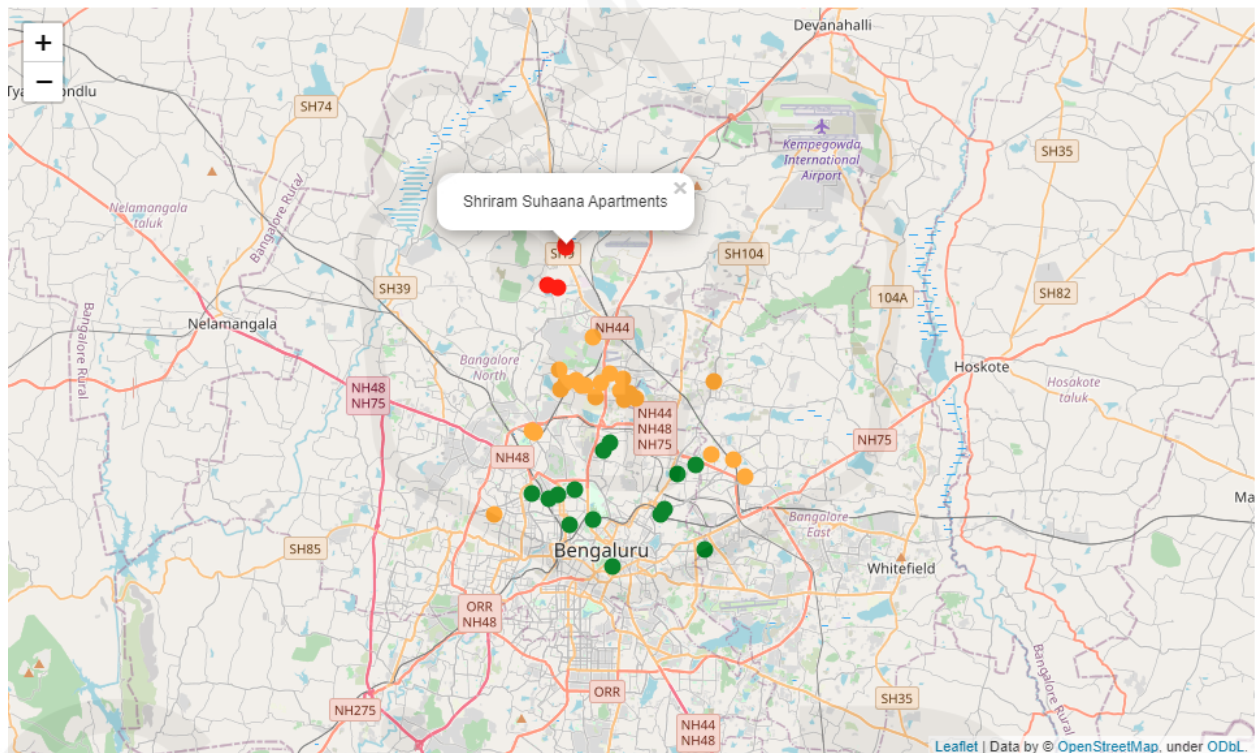
The project consists of the following stages:



## High-Level Approach

- Fetch Datasets from the relevant locations (Data Collection)
- Clean the Datasets to prepare them for analysis. (Data Cleaning via Pandas)
- Visualise the data using boxplots. (Using Matplotlib /Seaborn /Pandas)
- Fetch Geolocation Data from the Foursquare API. (REST APIs)
- Use K-Means Clustering to cluster the locations (Using ScikitLearn)
- Present findings on a map. (Using Folium/Seaborn)

The desired end result of this project is something like this:



## Applications

K-Means clustering is used in a variety of examples or business cases in real life, like:

- Academic performance (grouping students by their learning rate)
- Diagnostic systems (grouping system faults under various reasons)
- Search engines (grouping search results)
- Wireless sensor networks (Mapping networks)

The FourSquare API data can be used for:



- Building a restaurant review app like Swiggy Zomato etc.
- Supporting a ride sharing service like Uber Pool

## Task 1

### Getting Started

We need data to do data analysis! Fetch the data we need and set up your environment before you move on to data analysis.

### Requirements

- Head over to [this](#) link to get a dataset we'll be using. If you run into "too many requests", here's a [drive link](#).
- Note that you'll need both files, the csv and the document - the document explains the meaning of the CSV values.
- You will need a [jupyter notebook](#) installed in your environment.
- Explore the codebook\_food file. There are around 70 parameters. Not all of them are relevant. Think carefully about which ones are the most useful. Which ones can be used to quantifiably differentiate students? A good example of this is income. A more qualitative parameter like "How much they like vegetables" on a 1-5 scale might not be as useful.
- Extract the most relevant features into a pandas dataframe.
- This process of Extracting the features, (and dealing with different kinds of values as well as NaN values) is known as **Data Cleaning**.

### References

- [Reading CSV Manual Page](#)
- [How to load CSV into Jupyter](#)
- [Data Cleaning - Methodology/Approach](#)
- [Data Cleaning - Useful Tools](#)

### Tips

- Jupyter can be run on any OS, but for Windows, [Anaconda](#) provides the most hassle-free way of installing Jupyter.
- You may want to use [loc](#) when doing data cleaning.
- Non numeric fields are much harder to deal with effectively as compared to numeric fields.

### Expected Outcome

You should have a pandas dataframe with the relevant parameters ready to go for analysis in a jupyter notebook. Something like this will do:



	cook	eating_out	employment	ethnic_food	exercise	fruit_day	income	on_off_campus	pay_meal_out	sports	veggies_day
0	2.0	3	3.0	1	1.0	5	5.0	1.0	2	1.0	5
1	3.0	2	2.0	4	1.0	4	4.0	1.0	4	1.0	4
2	1.0	2	3.0	5	2.0	5	6.0	2.0	3	2.0	5
3	2.0	2	3.0	5	3.0	4	6.0	1.0	2	2.0	3
4	1.0	2	2.0	4	1.0	4	6.0	1.0	4	1.0	4

## Task 2

### Data Exploration and Visualisation

Now that we have our data, we need to understand it. A good way to do this is by visualising the data via graphs. Graphs help us quickly get a sense of the data, and are a much more user-friendly way of understanding data as compared to reading thousands of rows of data!

A good graph to look at distributed groups is a [Boxplot](#). It can tell us at glance where the population is concentrated, and how the outliers compare to the average object in the group.

### Requirements

- To visualise the `food_choices` dataset you created in the previous task, plot a [boxplot](#) on the dataframe.
- Refer to the `codebook_food` file to make sense of the Boxplot. Note down the general trends followed by the population that was surveyed as part of the study.
- You should be looking at things like how much each person exercises on average (these people will need gyms), whether they are vegetarian / non vegetarian (we should account for both), and income (this affects the lifestyle of the person significantly.)

### Bring it On!

Box plots are just one way to visualise your data. Can you think of any other visualisation methods that might aid your analysis?

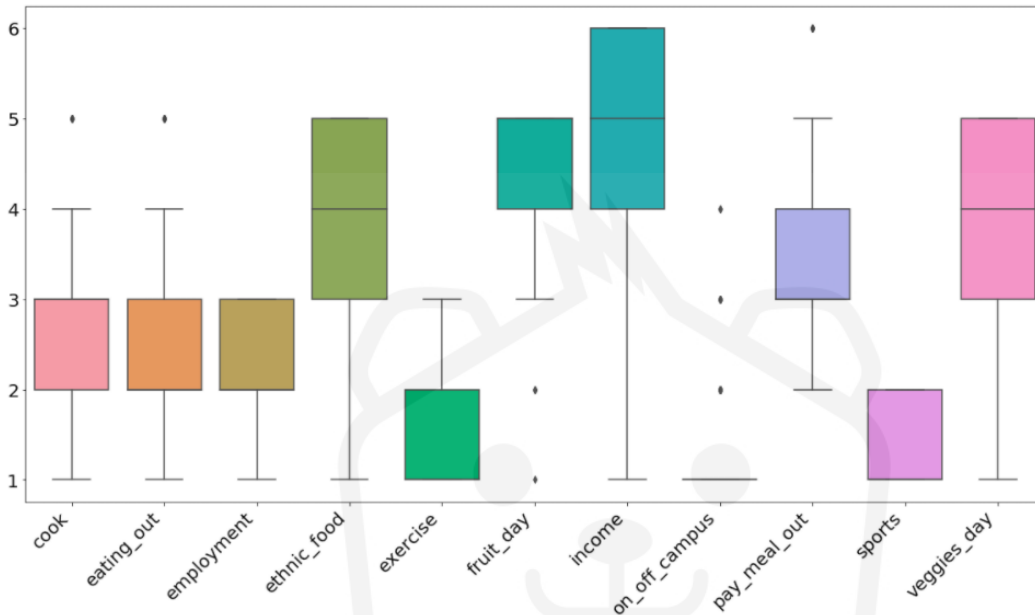
### References

- [Understanding Box Plots](#)
- [Boxplots using Seaborn](#)
- [Boxplots using Pandas](#)



## Expected Outcome

You should have a boxplot of your cleaned dataset, and a list of insights pertaining to the dataset.



## Task 3

### Run KMeans Clustering on the data

K Means Clustering will help us group locations based on the amenities located around them. For example, a location with a high amount of shops nearby will be labeled "Amenity Rich" while a location with less amenities will be labeled "Amenity Poor". Similar locations will be grouped (clustered) together. (Check the references for a more formal explanation!)

Run the KMeans Clustering Algorithm and figure out the best value for K, which we will use later.

Note that here we are applying K means first on the dataset of the general population, which will help us organise the population into groups. Further down the line, we will apply K means again on a different dataset.



## Requirements

- Run [KMeans clustering](#) on the dataset you have. Use trial and error to figure out the best value of K. (Note: The best value of K is where the clusters are clearly demarcated on particular attributes, e.g. income)
- Note the difference in the clusters as you step through different values of K. It might be a good idea to plot boxplots again to see if there's any visible demarcation based on different parameters.
- Once you are settled on a K value, preserve the boxplots and jot down any insights you find. We'll need these later!

## Tip

Income is usually a reliable way to differentiate people, as different classes of people tend to have different habits.

## References

- [K means Clustering](#)
- [Finding an Optimum K Value](#)
- [Boxplots using Seaborn](#)
- [Boxplots using Pandas](#)

## Expected Outcome

You should have the optimum K value for the dataset, and the parameters on which the clusters are differentiated.

## Task 4

### Get Geolocational Data from Foursquare API

Now that we know the best K value for our population, we need to get geolocational data from the Foursquare API to find these people some accommodation!

## Requirements

- Make a free [foursquare account](#) and get your API credentials set up. (Note that there are limits on a free account, so be careful about calling the API!)
- Set up your query in such a way that you can check for residential locations in a fixed radius around a point of your choosing. For example, you can pick (13.133521,77.567135) if you want a sample location in Bangalore. Here's how the API response might look:





	categories	hasPerk	id	location.address	location.cc	location.city	location.country	location.crossStreet	location
0	'4d954b06a243a5684965b473', 'name': 'R...'	False	4db7040e0437fa536a641766	Banaswadi Main Rd	IN	Bangalore	India	hight street	
1	'4d954b06a243a5684965b473', 'name': 'R...'	False	594f23a82be42528bc56a739	NaN	IN	Bangalore	India	NaN	
2	'4d954b06a243a5684965b473', 'name': 'R...'	False	51319d59e4b04a7c6799abe4	Ananthapura Road	IN	Bangalore	India	Yelahanka New Town	
3	'4d954b06a243a5684965b473', 'name': 'R...'	False	56133261498e95c619c830f8	NaN	IN	Bangalore	India	NaN	
4	'4bf58dd8d48988d12b951735', 'name': 'B...'	False	52dd2fd2498ebd1fc2edf286	NaN	IN	NaN	India	NaN	

- Hit the endpoint, and parse the response data into a usable dataframe. There is a lot of information you don't need, so apply the same data cleaning principles you used in Task 1 to get a workable dataframe.
- We also need a count of grocery stores, restaurants, gyms etc. around each residential location. Form another query to get all these locations (fixed in a short distance around each residential location) and hit the endpoint again. Here's how this API response might look:

```
Out[21]: [{'id': '4ebf572e77c885a64e41ca74',  
  'name': 'Om Ganesh Fruit Juice Centre',  
  'location': {'lat': 12.988079790170037,  
    'lng': 77.61756356265425,  
    'labeledLatLngs': [{'label': 'display',  
      'lat': 12.988079790170037,  
      'lng': 77.61756356265425}],  
    'distance': 2489,  
    'cc': 'IN',  
    'city': 'Bangalore',  
    'state': 'Karnātaka',  
    'country': 'India',  
    'formattedAddress': ['Bangalore', 'Karnātaka', 'India']},  
  'categories': [{'id': '4bf58dd8d48988d112941735',  
    'name': 'Juice Bar',  
    'pluralName': 'Juice Bars',  
    'shortName': 'Juice Bar',  
    'icon': {'prefix': 'https://ss3.4sqi.net/img/categories_v2/food/juicebar_',  
      'suffix': '.png'}.}
```

- Clean up the data in the same way as before - drop the irrelevant values, handle the NaN values(if any) and summarise the results into a dataframe. [Click here](#) for a refresher if you need it.





## Tips

- NaN values are okay, as long as you don't have too many in the dataframe. Pay attention to selecting the parameters you need first, before dealing with other aspects of Data Cleaning.
- You will want to filter locations by distance. A shop 50 km away from your house isn't exactly useful!

## References

- [Foursquare API Docs](#)
- [Search Endpoint Docs](#)
- [What is a REST API?](#)
- [Using Postman to Test API endpoints](#)

## Expected Outcome

You should have a dataframe with the locations (in latitude, longitude) format, along with the counts of how many amenities are present around each location.

	lat	lng	Restaurants	Fruits,Vegetables,Groceries
0	13.000170	77.624545	50	49
1	13.113421	77.568541	23	7
2	13.114882	77.563090	12	4
3	13.039492	77.555590	50	29
5	13.135727	77.572482	13	3
6	13.134563	77.572646	13	3
7	12.979005	77.645620	49	49
8	13.069250	77.595719	49	19
9	13.064801	77.572098	50	18
10	13.063300	77.580675	50	20

## Task 5

### Plot the clustered locations on a map

Now it's time to run K Means clustering on the data (finally!) and plot the results on a map.

Note that here we are applying K means on the dataset of the locations which we chose, which will help us find the best location for each population group that we found in Task 3.

### Requirements

- Run K Means clustering on the dataset you prepared in the previous milestone, with the optimal K value you found.
- Now that you have the results, it's time to visualise them. Using [Folium](#), Plot your results on a map of the world, centered on the location you chose.
- Define a proper colour scheme so the locations are easily differentiated by the cluster number.

### Bring it On!

Now that you have the results, try and make sense of them - what sort of person will live in what cluster? Are there any particular criteria/demographics that go well together? Knowing all this will help you present your findings if you were doing a case study as a Data Scientist!

### References

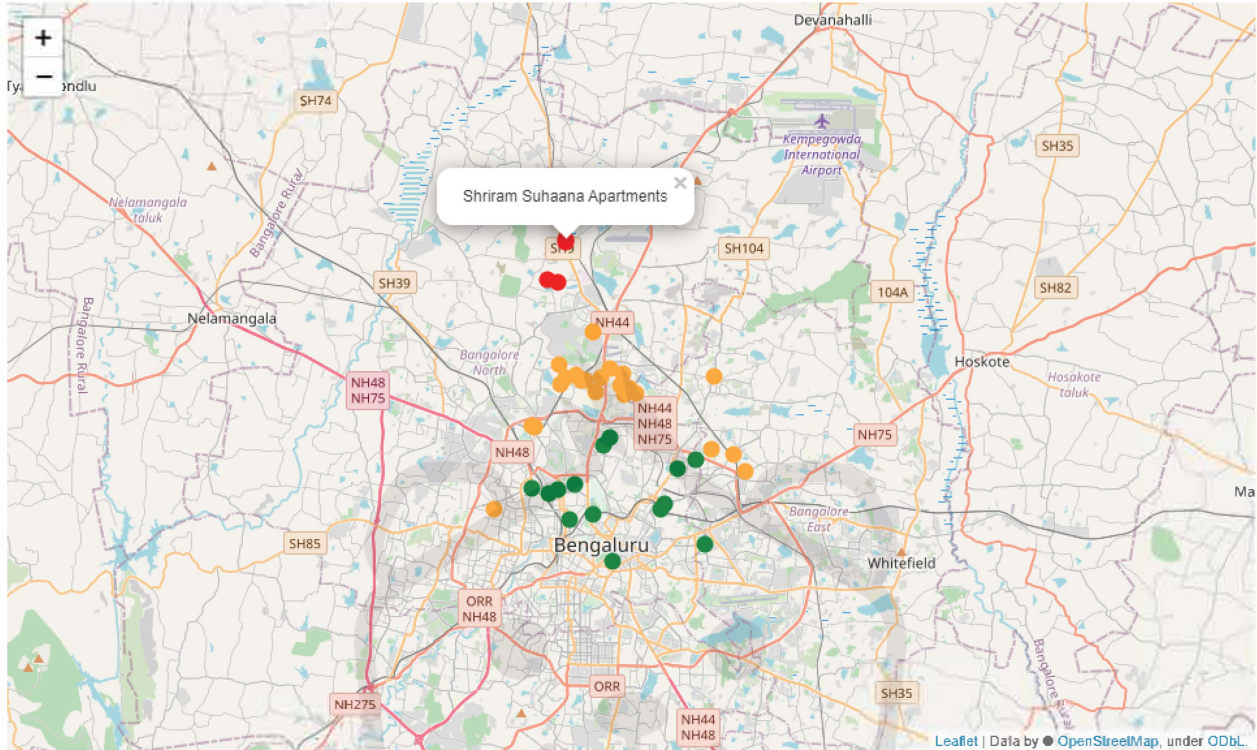
- [K Means clustering \(SKLearn\)](#)
- [Folium Documentation](#)

### Expected Outcome

You should have a map centered on the location you chose, with the locations differentiated by colour scheme. Have a look at the screenshot below for reference.



Crio.Do



Crio.Do