**REPORT**

**Capstone Project - The Battle of Neighbourhoods**

**(Finding nearby places for tourist in Mumbai)**

**INTRODUCTION:**

[**Tourism**](https://en.wikipedia.org/wiki/Tourism_in_India)**in**[**Mumbai**](https://en.wikipedia.org/wiki/Mumbai) is an industry that attracts almost 6 million tourists per year, making it the 30th-most visited location worldwide. According to [United Nations](https://en.wikipedia.org/wiki/United_Nations), as of 2018, Mumbai was the second most populous city in India after Delhi and the seventh most populous city in the world with a population of 19.98 million.

Mumbai offers natural heritage and modern entertainment including leisure spots, beaches, cinemas, studios, holy places, amusement parks and historical monuments. Transport options include air, road, train and ship.

As tourists would be interested in exploring places they would need a list of places and an itinerary. In order to make it easier they can use this program to identify all the places of interest in no time and it would be reliable as the places are located nearby. Thus, this program will help to find the underlisted places, nearby:

1. Restaurants
2. Parks
3. Hotels
4. Cafeteria
5. Shopping

PROBLEMS:

To find the answers to the following questions

1. List and visualise all major restaurants in Mumbai.
2. List and visualise all major Parks in Mumbai
3. List and visualise all major Hotels in Mumbai
4. List and visualise all major Cafeteria in Mumbai
5. List and visualise all major Shopping in Mumbai

**DATA DESCRIPTION**

For this project we need the following data:

1. Restaurants in Mumbai (India)
   1. Data source: Foursquare API
   2. Description: By using API we will get all restaurants in neighbourhood. We can filter these restaurants to only get nearby restaurants.
2. Parks in Mumbai (India)

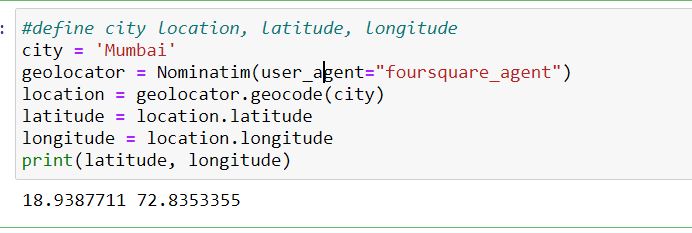
2.1 Data source: Foursquare API

2.2 Description: By using API we will get all hotels in neighbourhood. We can filter these restaurants to only get nearby parks.

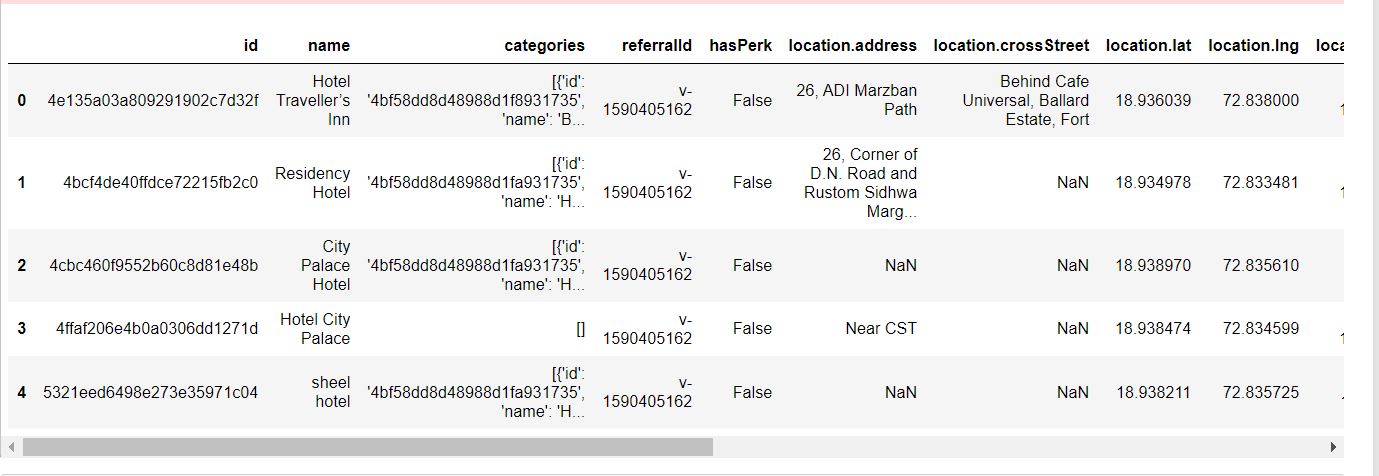
1. Hotels in Mumbai (India)
   1. Data source: Foursquare API
   2. Description: By using API we will get all hotels in neighbourhood. We can filter these restaurants to only get nearby hotels.
2. Cafeteria in Mumbai (India)
   1. Data source: Foursquare API
   2. Description: By using API we will get all cafeteria in neighbourhood. We can filter these restaurants to only get nearby cafeteria.
3. Shopping in Mumbai (India)
   1. Data source: Foursquare API
   2. Description: By using API we will get all shopping in neighbourhood. We can filter these restaurants to only get nearby shopping.

**METHODOLOGY**

1. We begin by searching the city, its longitude and latitude:



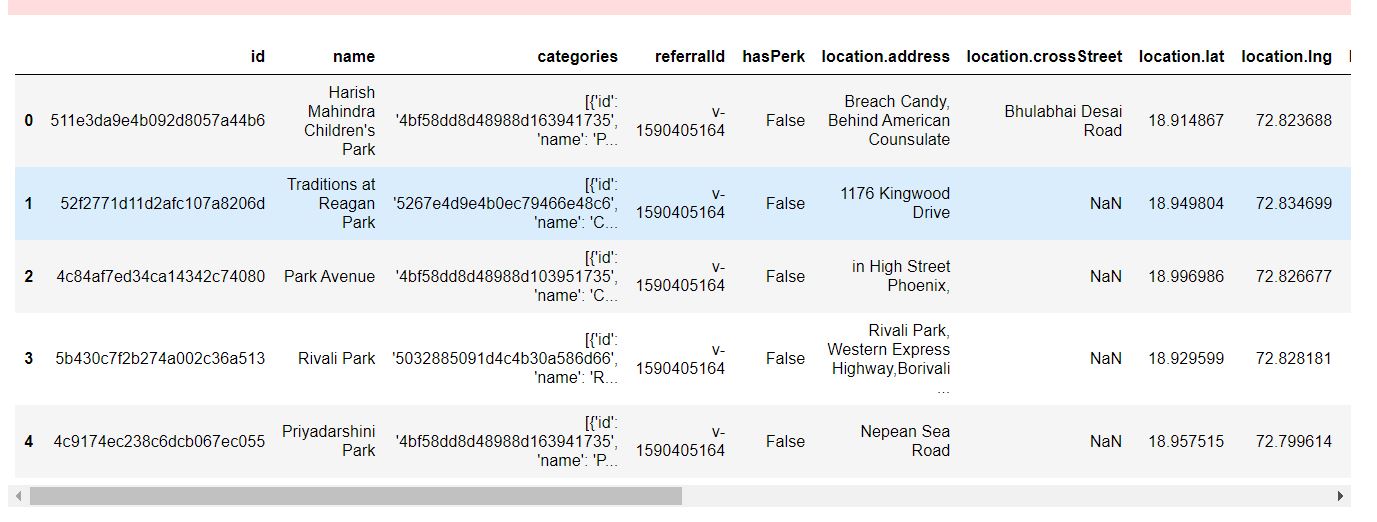
1. Then searching the Hotels from the searched city:



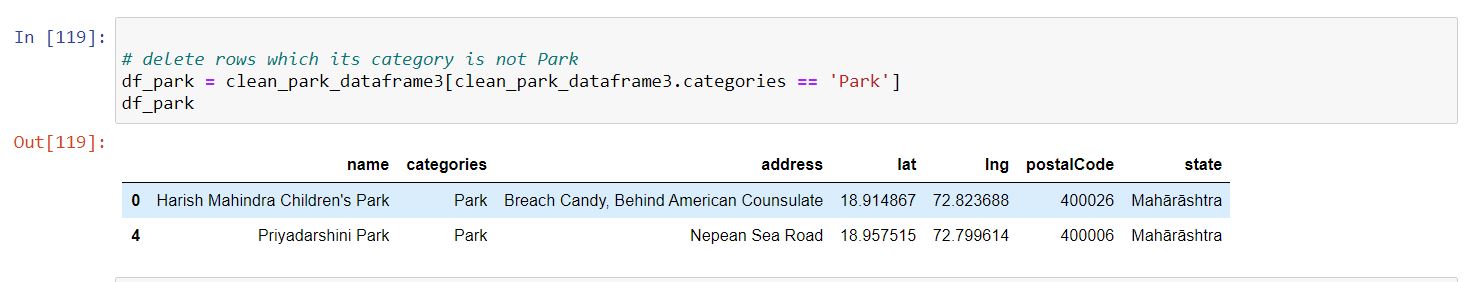
1. Now cleaning the search result ie. Deleting unnecessary columns, deleting rows with none values, deleting rows which its category is not Hotel, deleting rows which has duplicate hotel’s name.



1. Now searching the park in Mumbai:



1. Cleaning the collected park dataframe:



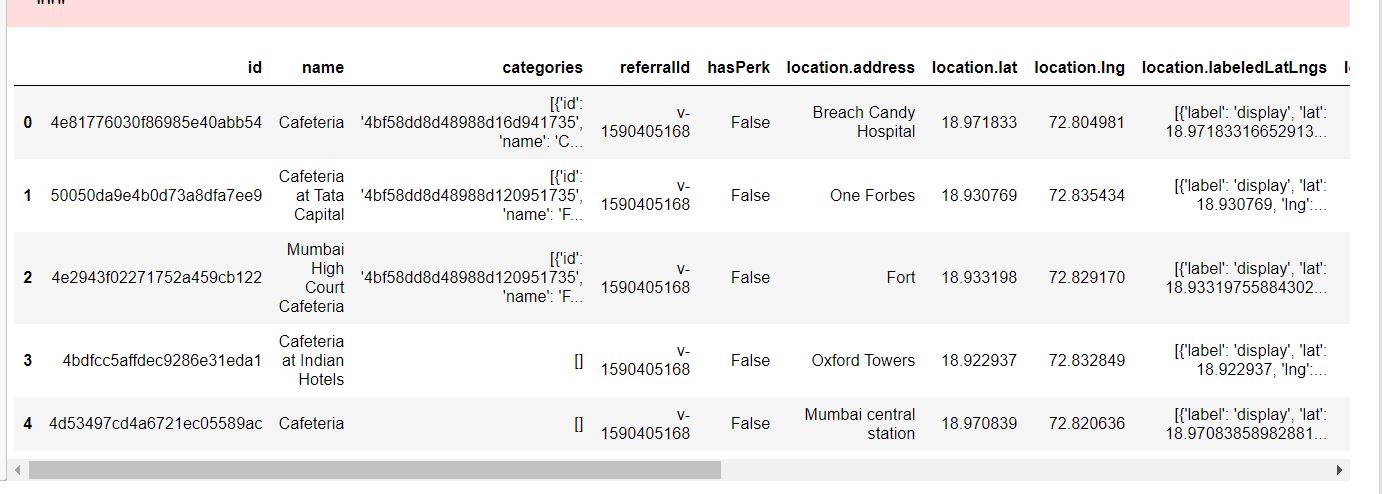
1. Now searching the nearby restaurants in Mumbai:



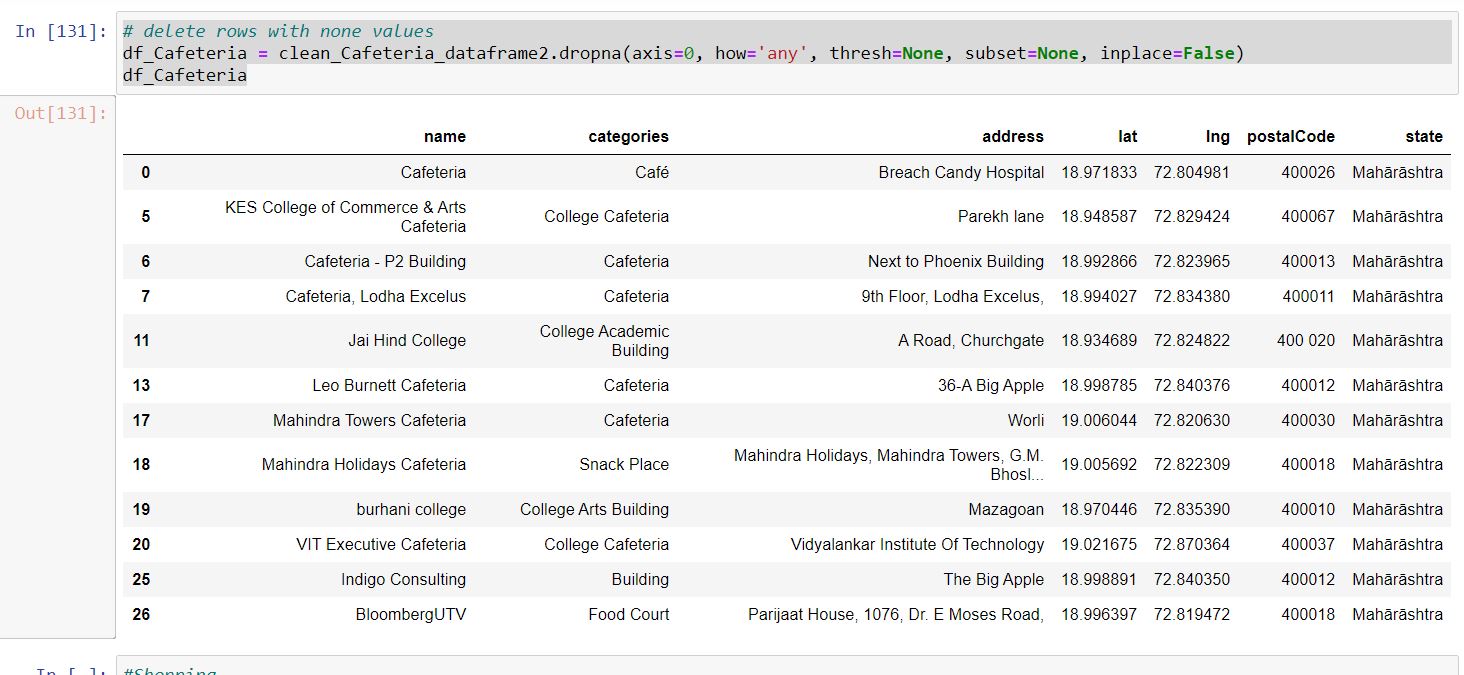
1. Cleaning the collected restaurant dataframe:



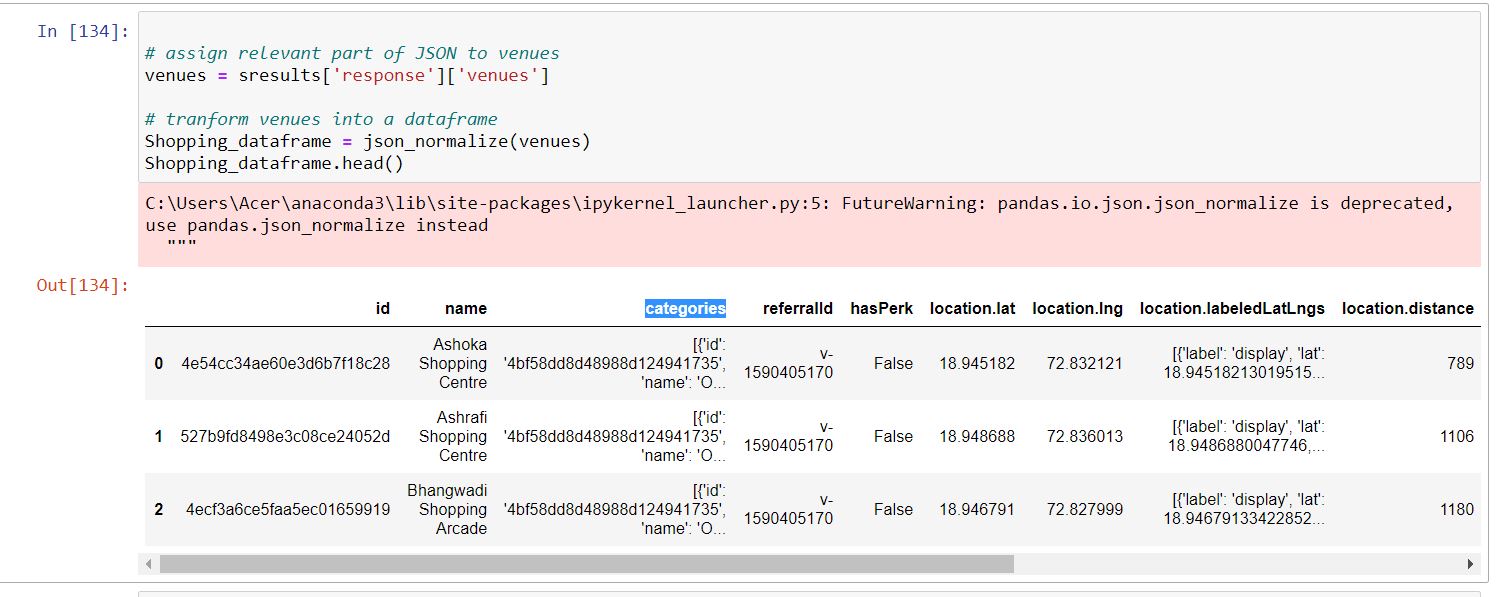
1. Now collecting the nearby cafeteria in Mumbai:



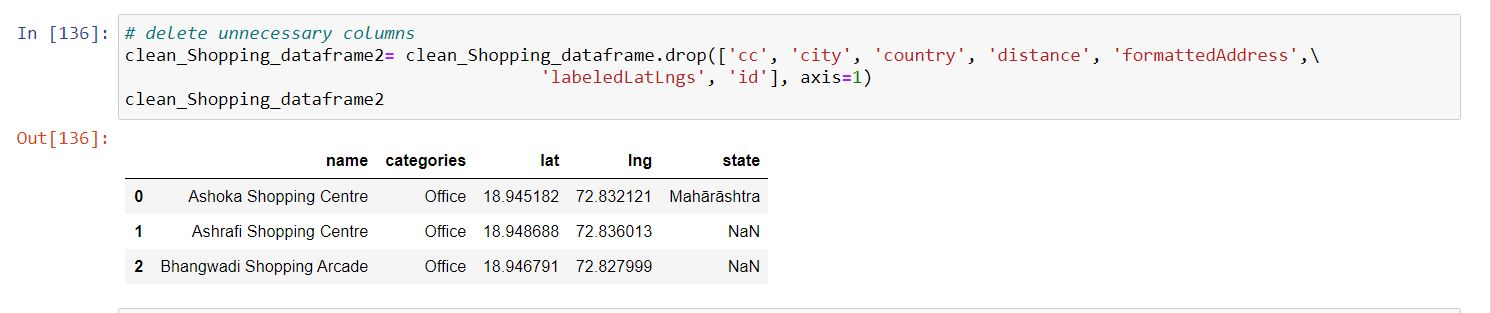
1. Cleaning the cafeteria dataframe:



1. Now collecting the nearby shopping:



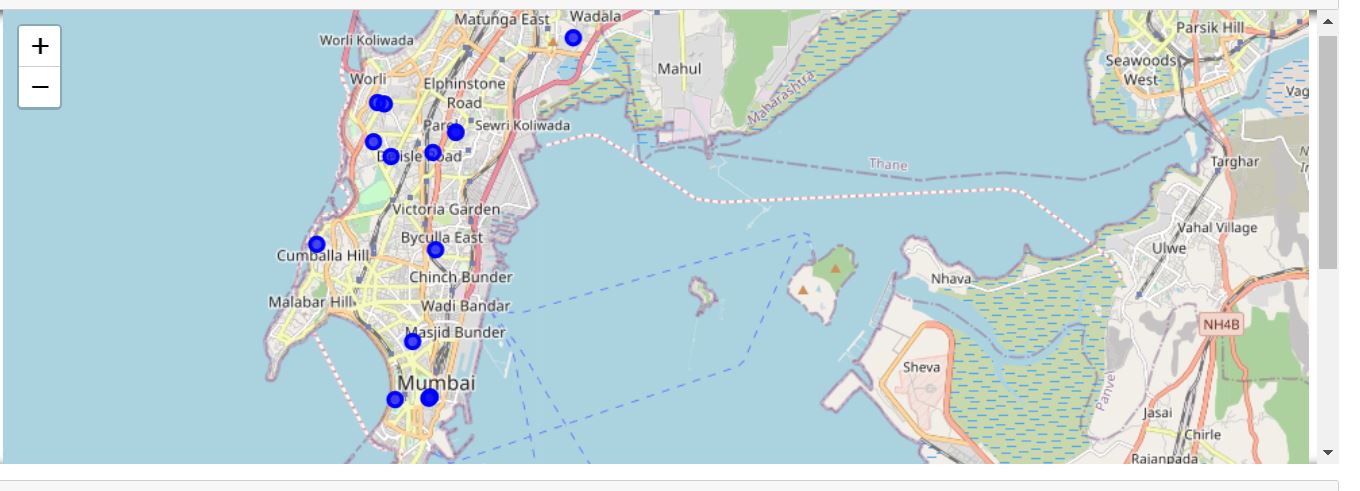
1. Now cleaning the shopping dataframe:



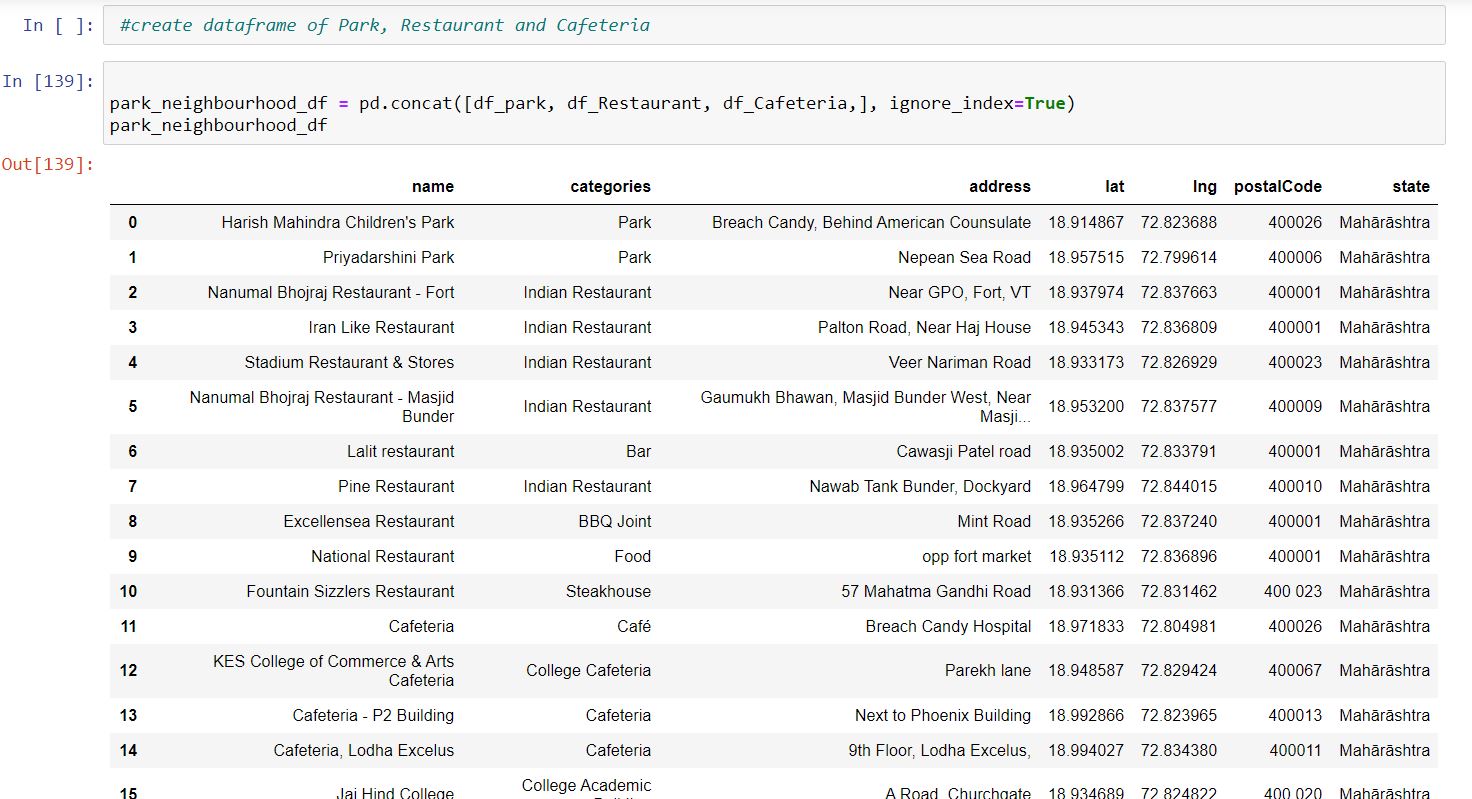
1. Now creating the dataframe of hotels, shopping stores and cafeteria



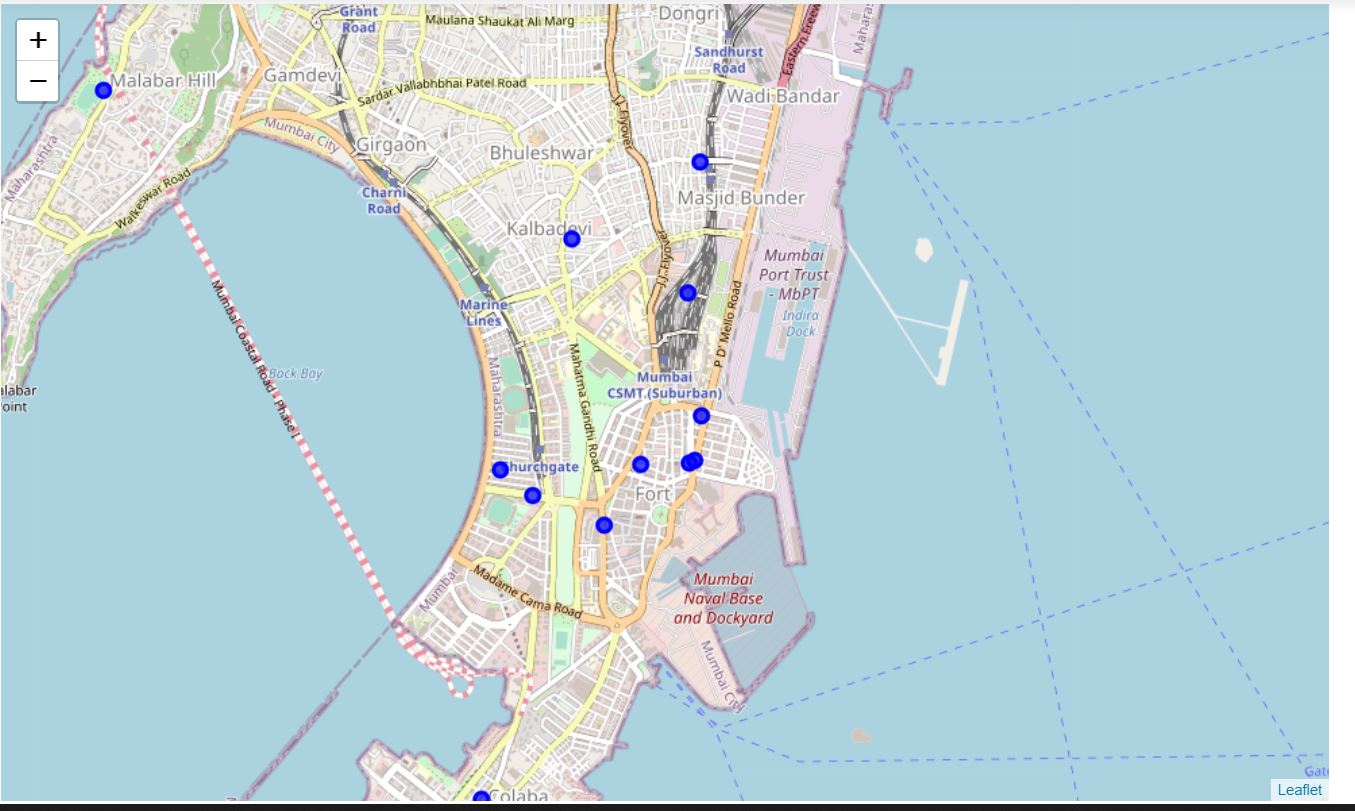
1. Visualization the above dataframe ie. Of hotels, shopping and cafeteria:



1. Making dataframe of park,restaurant and cafeteria:



1. Visualization of dataframe of park,restaurant and cafeteria:



**RESULT:**

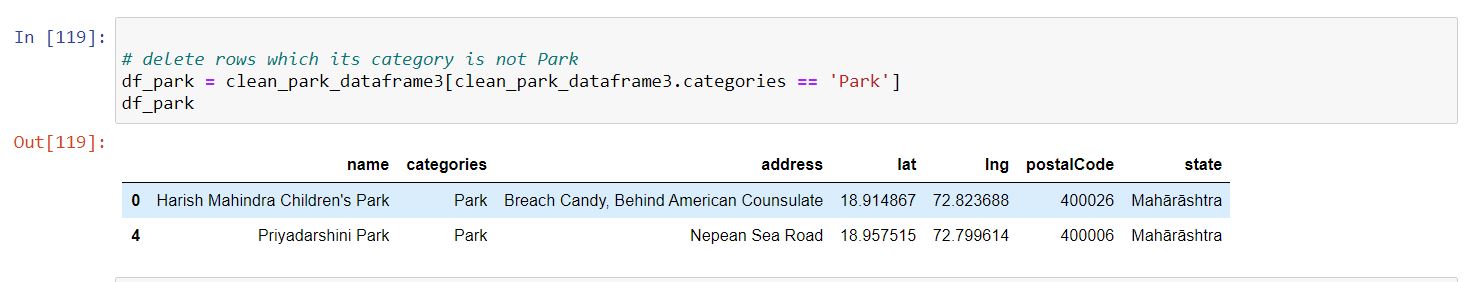
So now we can answer the questions asked above in Question section:

Answers:

1. The nearest Hotel are:



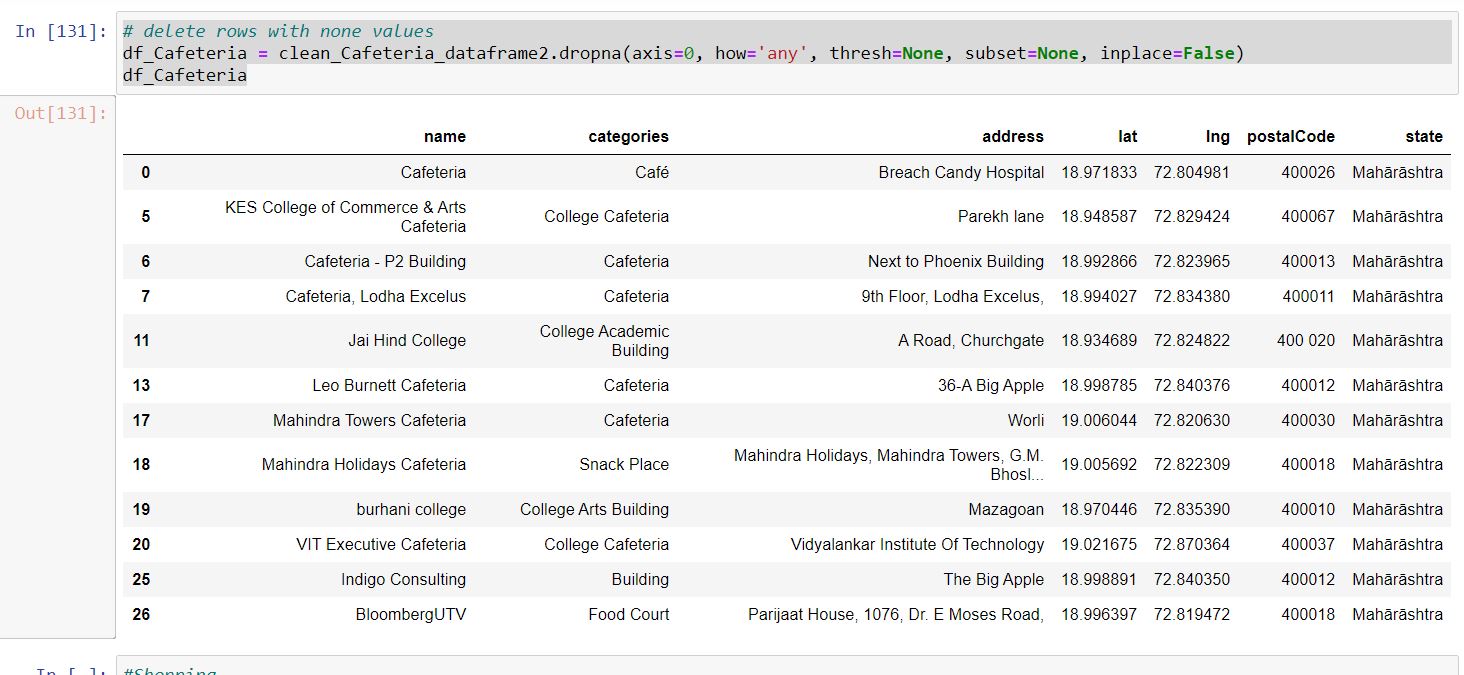
1. The nearest park are:



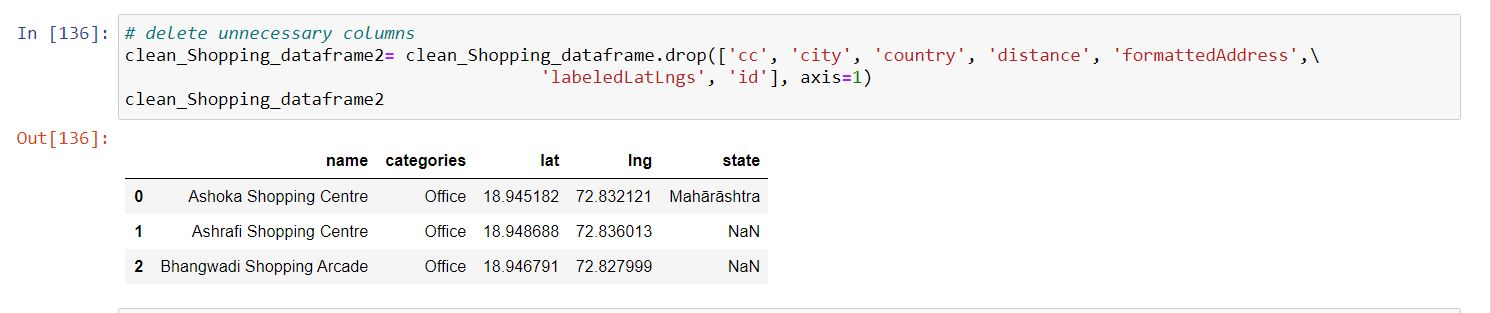
1. The nearest restaurant are:



1. The nearest Cafeteria are:



1. The nearest shopping center are:



CONCLUSION

There is always room for improvement and hence the above solution I have provided can also be improved for best results depending upon the data we have.