Analysis and Visualization of COVID19 in India at District Level and Availability of Health Care at regional level in Hyderabad, India.

Saquib Mohiuddin Siddiqui
Capstone Project
IBM Data Science Professional Certificate

1. Introduction

1.1 Background

Coronavirus disease 2019 (**COVID-19**) is an infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). It was first identified in December 2019 in Wuhan, Hubei, China, and has resulted in an ongoing pandemic the first confirmed case has been traced back to 17 November 2019 in Hubei As of 27 July 2020, more than 16.2 million cases have been reported across 188 countries and territories, resulting in more than 648,000 deaths. More than 9.4 million people have recovered.

COVID-19 is a new disease, and many of the details of its spread are still under investigation. It spreads easily between people—easier than influenza but not as easily as measles.

1.2 Problem

India is the second most populous country with a population of over 1.3 Billion, and a population density of 700 per sq. km and boasts an incredibly diverse culture. Hence, it is at one of the highest risk of community transmission and uncontainable spread of the disease so it is compelling to have quality and adequate health care facilities in place to provide necessary health care sufficiently to all. In this Project we will analyze and visualize the transmission and deaths caused by the virus in India using open source data.

Followed by which, using **Four-square API** we will analyze and visualize the health care facilities available to combat the virus spread across all localities in my hometown Hyderabad, India.

The Data obtained can be used to implement better health care measures and also identify hotspots to isolate them to mitigate the pandemic spread.

2. Data

Data Acquisition and Wrangling

I have used two different data sets

For the first data set, the Data has been acquired from open source website https://www.covid19india.org/ using their API's. https://api.covid19india.org/documentation/csv/

The Data from the above source is used to get detailed district wise data for India

For the second data set, the data for all neighborhoods in Hyderabad have been acquired via Web Scraping using BeautifulSoup, I've referred to Wikipedia https://en.wikipedia.org/wiki/Category:Neighbourhoods_in_Hyderabad,_India to get the data for all neighborhoods in Hyderabad, and appended the missing localities with a simple *list.append* function.

Deleted Columns	Retained Columns
Migrated_Other	District
Delta_Confirmed	State
Delta_Active	Confirmed
Delta_Recovered	Active
Delta_Deceased	Recovered
District_Notes	Deceased
Last_Updated	
District_Key	

Methodology

The Project will utilize the following libraries:

- Pandas and Numpy – for data manipulation and analysis
- BeautifulSoup – for Scraping Web Data
- FourSquare API – to get district level data.
- Matplotlib – to plot various graphs
- Plotly – to plot interactive graphs
- Folium – To render maps
- Geocoder – to get location coordinates

We start off by installing all the necessary libraries. The coordinates for the Hyderabad city is obtained using geocoder followed by which a map is rendered using folium. It is done by creating a data frame "*hyderabad_map*", and loading the coordinates, we converted the variable of coordinate's to float, incase if the returned value is in the form of a string. This gives the visual of the city we are going to work with. **Refer to Fig (1) in Results section for the rendered image.**

The Second step involves acquiring the district level data of India from the source listed above, the data is loaded from the url into data frame 'df' using pandas read function i.e., "pd.read_csv". Function. The raw data is then viewed as shown:

District	SlNo	State_Code	State	District_Key	Confirmed	Active	Recovered	Deceased	Migrated_Other	Delta_Confirmed	Delta_Active	Delta_Re
Unassigned	0	UN	State Unassigned	UN_Unassigned	0	0	0	0	0	0	0	
Nicobars	1	AN	Andaman and Nicobar Islands	AN_Nicobars	0	0	0	0	0	0	0	
North and Middle Andaman	2	AN	Andaman and Nicobar Islands	AN_North and Middle Andaman	1	0	1	0	0	0	0	
South Andaman	3	AN	Andaman and Nicobar Islands	AN_South Andaman	51	19	32	0	0	0	0	
Foreign Evacuees	0	AP	Andhra Pradesh	AP_Foreign Evacuees	434	2	432	0	0	0	0	

Data wrangling is done by dropping unnecessary data, this is done by simple *del* functions on data frame. There is also the data which can return non feasible results, hence those fields are remove too by passing a conditional parameters to the pandas data frame. The following is the result of cleaned data and loaded into data frame *df_final:*

District	State_Code	State	Confirmed	Active	Recovered	Deceased
Unassigned	UN	State Unassigned	0	0	0	0
Nicobars	AN	Andaman and Nicobar Islands	0	0	0	0
North and Middle Andaman	AN	Andaman and Nicobar Islands	1	0	1	0
South Andaman	AN	Andaman and Nicobar Islands	51	19	32	0
Foreign Evacuees	AP	Andhra Pradesh	434	2	432	0
...	...	...	...	...	...	...
Purba Bardhaman	WB	West Bengal	633	256	373	4
Purba Medinipur	WB	West Bengal	1305	558	736	11
Purulia	WB	West Bengal	174	52	122	0
South 24 Parganas	WB	West Bengal	4442	1469	2900	73
Uttar Dinajpur	WB	West Bengal	863	324	533	6

762 rows × 6 columns

Followed by which, the bar charts are plotted using matplotlib function where a mean of the data is passed and groupby is used to get statistical data.

The third step involves the analysis of district level data for Telangana State, this is done by using .loc() command in pandas which is used to access a group of rows or columns; here I used the Boolean operator "==" to get the data specific to Telangana. Followed by, using matplotlib to plot the bar charts. Where Hyderabad district reports the highest cases, the bar and pie charts are plotted to show the cumulative numbers and shares respectively.

District	State_Code	State	Confirmed	Active	Recovered	Deceased
Foreign Evacuees	TG	Telangana	33	33	0	0
Other State	TG	Telangana	250	250	0	0
Adilabad	TG	Telangana	177	162	15	0
Bhadradri Kothagudem	TG	Telangana	156	152	4	0
Hyderabad	TG	Telangana	35970	35642	305	23

The fourth step involves the analysis of neighborhood level data, by first acquiring the district data via web scraping using '*BeautifulSoup Library'*. I loaded the data from the source mentioned above into the data frame and performed wrangling by slicing the irrelevant data and adding the missing data using *list.append* function. We get the shape of 220, 1 which represents 220 localities in total. The geocoding library is once again called to get the coordinates for all the neighborhoods obtained. Below is the clean data obtained?

	Neighbourhood	Latitude	Longitude
0	A. S. Rao Nagar	17.411200	78.50824
1	A.C. Guards	17.393001	78.45690
2	Abhyudaya Nagar	17.337650	78.56414
3	Abids	17.389800	78.47658
4	Adibatla	17.235790	78.54132
...	...	...	...
215	Warsiguda	17.419350	78.51887
216	Yakutpura	17.360920	78.49083
217	Yapral	17.514290	78.52365
218	Yellareddyguda	17.433920	78.43468
219	Yousufguda	17.438350	78.42855

220 rows × 3 columns

I've obtained the list of hospitals within Hyderabad City using **FourSquare API.** This is done using *for* loop to get data for every locality obtained in the previous step and the corresponding values are added. Once the control comes out of loop we get the following data:

7]:

	Neighbourhood	Latitude	Longitude	Hospital	Hospital_Latitude	Hospital_Longitude	Hospital_category
0	A. S. Rao Nagar	17.41120	78.50824	Sowmya Hospital	17.408888	78.506439	Hospital
1	A. S. Rao Nagar	17.41120	78.50824	Andhra Mahila Sabha Hospital	17.402475	78.510062	Hospital
2	A. S. Rao Nagar	17.41120	78.50824	Dr. Kirans' Dental Hospital	17.403120	78.507499	Dentist's Office
3	A. S. Rao Nagar	17.41120	78.50824	Abhaya BBC New Born Children's hospital	17.401986	78.509492	Hospital
4	A. S. Rao Nagar	17.41120	78.50824	Sagarlal Memorial Hospital	17.415921	78.498460	Hospital
...	...	...	...	...	...	...	...
1412	Yousufguda	17.43835	78.42855	FehmiCare Hospital	17.436987	78.427228	Hospital
1413	Yousufguda	17.43835	78.42855	Susheela hospital	17.434595	78.435801	Hospital
1414	Yousufguda	17.43835	78.42855	Tanvir Hospital	17.429536	78.433841	Hospital
1415	Yousufguda	17.43835	78.42855	Lavanya Group Of Dental Hospitals	17.434936	78.436330	Dentist's Office
1416	Yousufguda	17.43835	78.42855	J J hospitals	17.444909	78.432509	Not Mentioned

1417 rows × 7 columns

Although the above data doesn't represent any unnecessary data, I used the *"dataframe.unique()"* function for *hospital category* to draw out the irrelevant data and following is the data I obtained:

Since we have obtained all the necessary Data, let us see the various categories to filter out unncessary data.

```
In [38]: 1 mydf['Hospital_category'].unique()

Out[38]: array(['Hospital', "Dentist's Office", 'Veterinarian', 'Not Mentioned',
                'Medical Center', 'Eye Doctor', 'Snack Place', 'Warehouse',
                'Conference Room', 'Train Station', 'Mental Health Office',
                'Medical School', "Doctor's Office", 'Mosque', 'Pharmacy',
                'Building', 'Hospital Ward', 'Metro Station', 'Bus Line',
                'Professional & Other Places', 'General Travel',
                'Indian Restaurant', 'College Academic Building', 'Optical Shop',
                'Ice Cream Shop', 'Road', 'Emergency Room',
                'South Indian Restaurant', 'Maternity Clinic'], dtype=object)
```

From above array, we can identify some irrelevant data crept in such as "Indian Restaurant", "Ice Cream Shop" etc.

The Data Wrangling is once again performed on the new Data Set, I've done it using simple Boolean operator ***"!="*** and loaded the data into a new data frame ***"mydf_new"***. The clean data is obtained, I verified it again using the "dataframe.unique" function, and the result obtained is:

```
In [40]:  1 mydf_new['Hospital_category'].unique() # Verifying all the irrelevant information is removed.

Out[40]: array(['Hospital', "Dentist's Office", 'Veterinarian', 'Not Mentioned',
                'Medical Center', 'Eye Doctor', 'Mental Health Office',
                'Medical School', "Doctor's Office", 'Pharmacy', 'Hospital Ward',
                'Emergency Room', 'Maternity Clinic'], dtype=object)
```

After obtaining clean data, we then get the coordinates for all the hospitals and a new data frame is created.

Finally, I analyzed the localities from the above data frame, by first obtaining the total hospitals within the city, the result is 174. I further plotted a bar graph with respect to different hospital categories. I then used *for* loop again with two input parameters, i.e. data frames obtained by web scraping and the data frame obtained with respect to hospitals and we get the data for localities that do not have any hospital within 1 Km radius. There are 46 such localities. The coordinates for those localities are obtained using *geocoder library*, where I used *'for'* loop to get the coordinates for every 46 localities. The following data was obtained:

	Neighbourhood	Latitude	Longitude
0	Adibatla	17.23579	78.54132
1	Aliabad, Hyderabad	17.34259	78.47626
2	Alwal	17.53543	78.54427
3	Ameenpur	17.53332	78.32529
4	Balapur, Ranga Reddy district	17.32281	78.50103

Then a map is rendered with pins of localities which have relatively low access to health care using folium library. This was done by first getting the coordinates range for the above data which is:

```
[17.31582000000003, 78.47764000000006]
```

The above coordinates are passed in the 'map' data frame and the *'for'* loop was used to plot and render the map with all the coordinates in the data frame.

We thus identified the regions which has localities with fewer health care facilities.

Results

Get a Map of Hyderabad using folium

The follium is a powerful interactive library to plot maps using coordinates

```
In [5]:
1 hyderabad_map = folium.Map(location = [float(HYD_Latitude), float(HYD_Longitude)], zoom_start = 12)
2 hyderabad_map.save("hyderabad_map.html")
3 hyderabad_map
```

Out[5]:

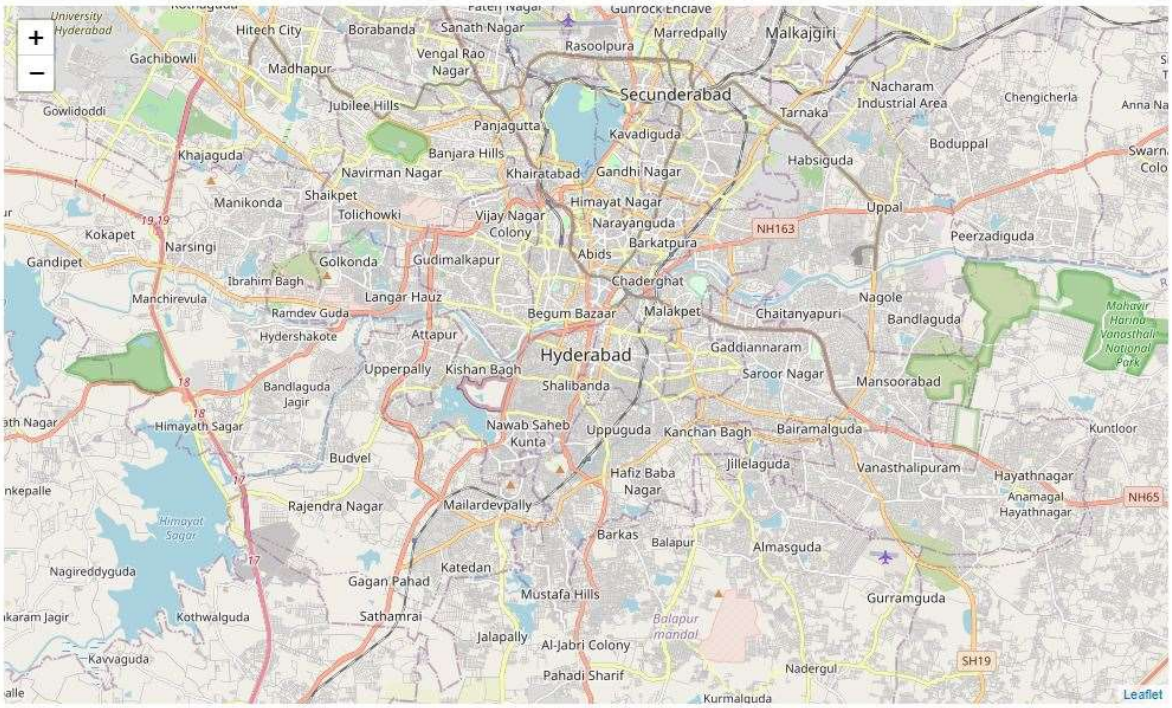

Fig (1): The Map of Hyderabad rendered using Folium

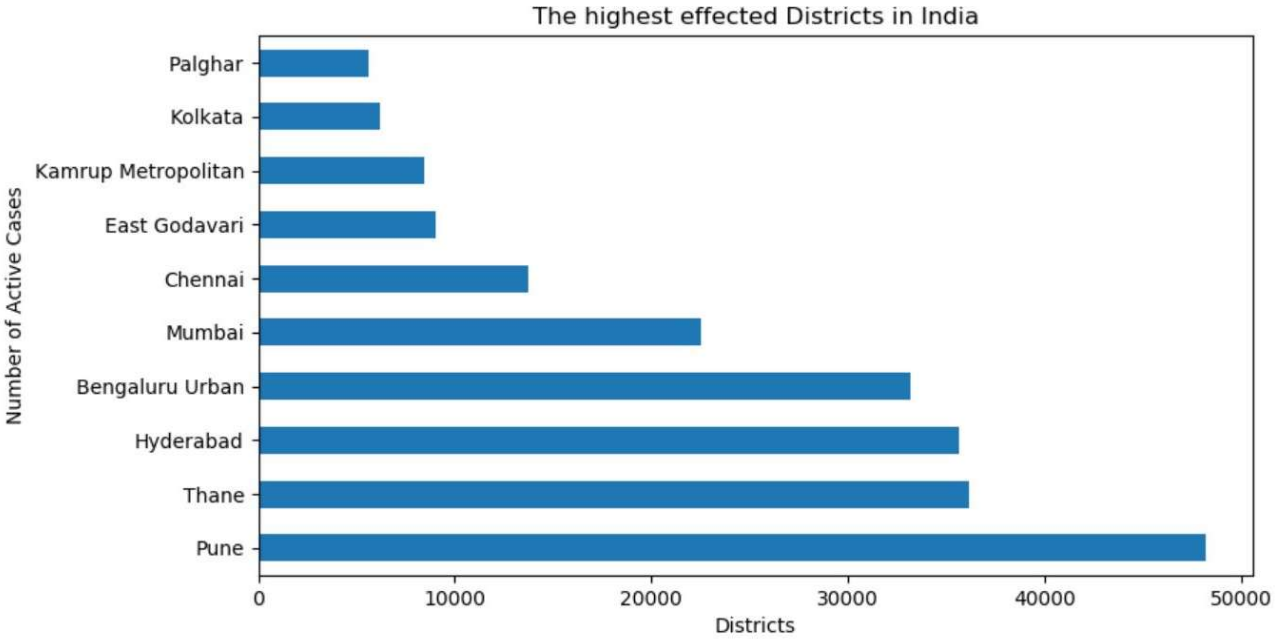

Fig (2): Bar Chart representing the most impacted districts in India

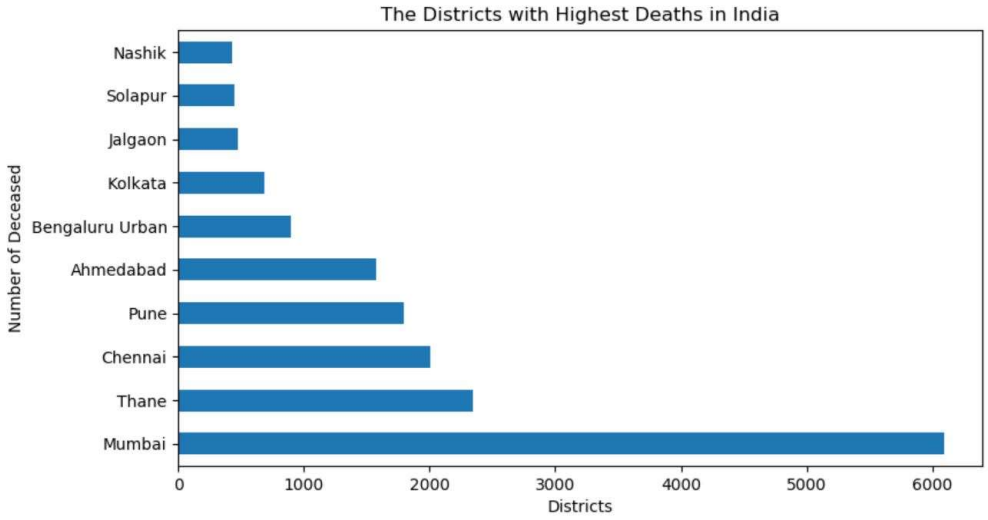

Fig (3): Bar Chart representing the districts with highest mortality rate in India

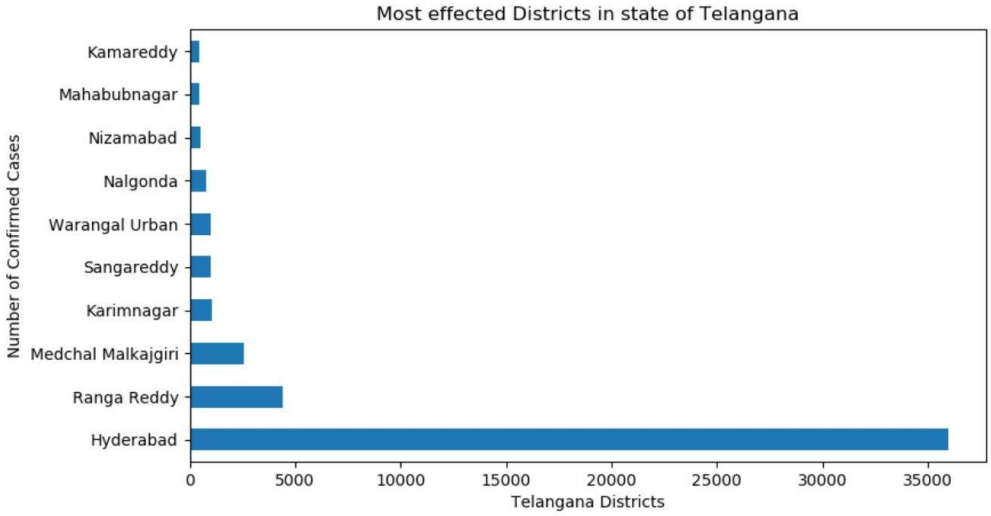

Fig (4): Bar Chart representing the most impacted districts in Telangana

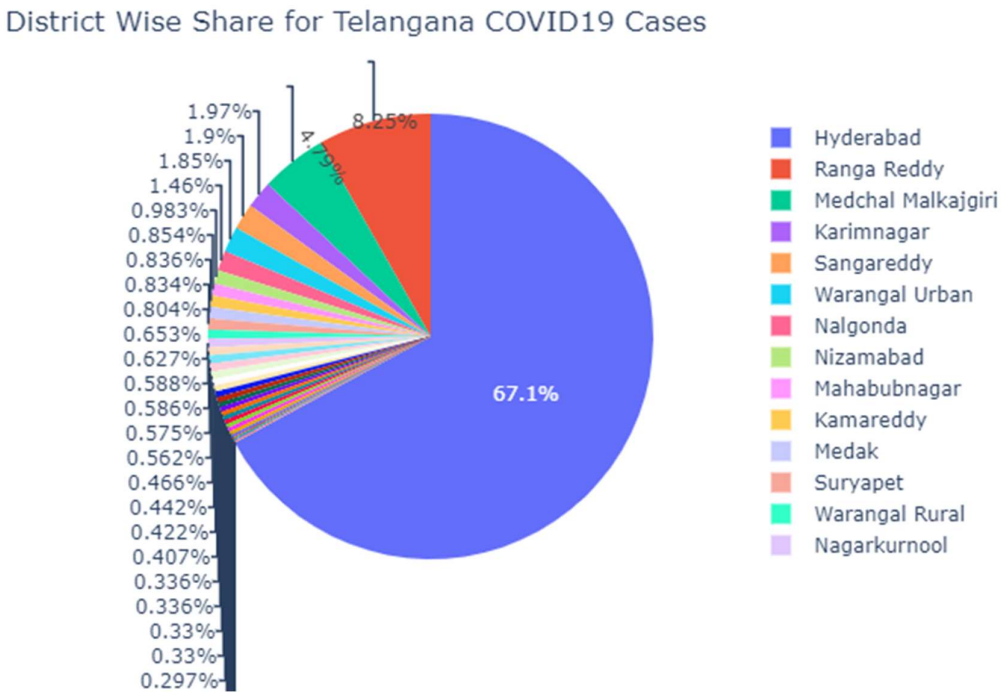

Fig (5): Pie Chart representing the district wise share of Telangana

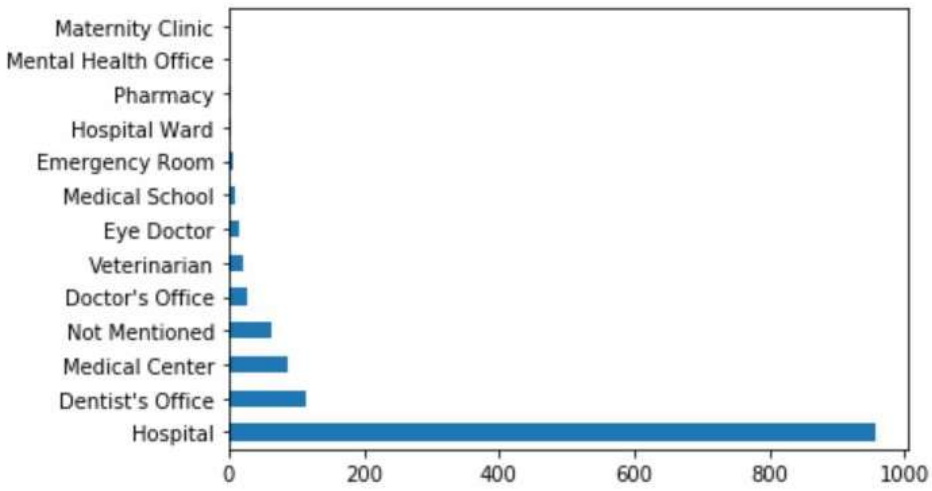

Fig (6): Categorical chart for Hospitals in Hyderabad

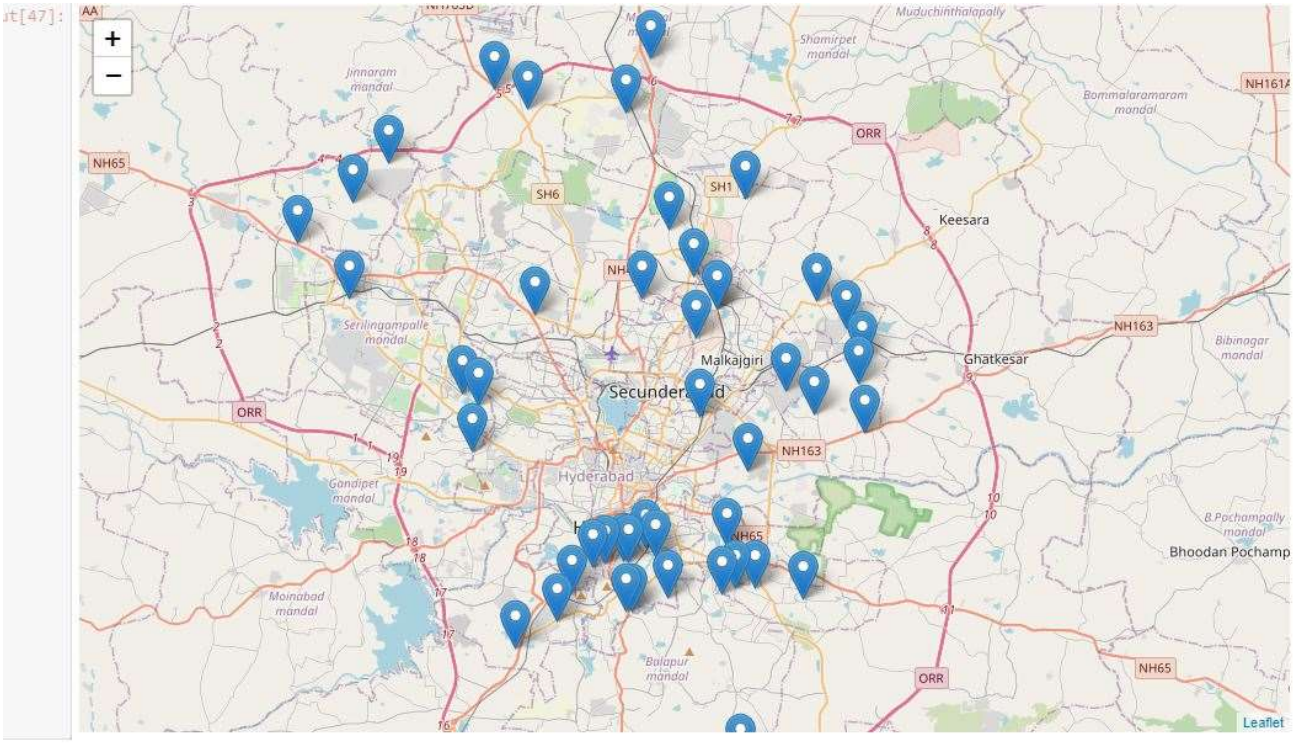

Fig (7): Rendered Map displaying the localities with no health care within 1 Km Radius.

Discussion

With population density as high as India, it becomes pivotal that all neighborhoods have access to health care facilities without having the hassle. Hence, The Primary objective of the project was to analyze the COVID19 spread in India and to present a visual data of localities in my hometown which requires the attention of authorities to plan and implement measures to avoid the community transmission and facilitate easier access to health care. From the rendered image above, we can identify the southern area (Old City) and the eastern area requires immediate attention as large clusters of neighborhood's are at risk.

Conclusion

In conclusion, after analyzing the different data sets, the project helps in better understanding of hotspots and neighborhood's which are at the relatively higher risk due to lack of facilities, thus, if implemented in real life could save millions living in the clusters above; the data presented isn't 100% accurate as the data isn't up to date due to state government not providing the recent data. However, The future of the project include further improvement if data be provided, by adding additional parameters such as number of hospital beds, ventilators available to analyze the data efficiently and enabling the authorities to plan and act accordingly. An ML model can also be developed to predict the capacities beforehand and build the necessary additional facilities required.