

Implementation 2: DBSCAN

In this section we want to use DBSCAN(Density-Based Spatial Clustering of Applications with Noise) algorithm to recognize dense covid-19 disease clusters in Iran. DBSCAN algorithm is a density based clustering algorithm which can perfectly cluster convex shapes of data sets and indicate the dense areas in our data based on two parameters: 1-Minpts which is representative of minimum number of points that a core point must have in a radius specified by 2-Epsilon.

Like previous section we **import** numpy, matplotlib.pyplot, and also DBSCAN from sklearn.cluster for pre prepared dbscan algorithm, and folium package which allows us to load a real world map and plot our data points on that map.

At first we read the csv file and keep it in numpy array as previous section. Then we plot data points with simple plot scatter as shown in figure below.

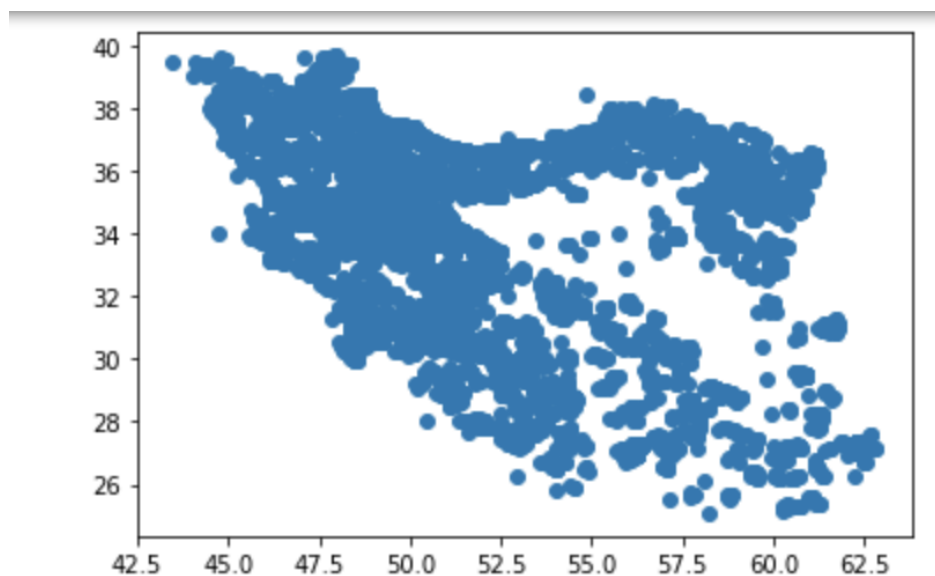


figure1- data points on simple plane

```
m = folium.Map(location=[32.427910, 53.688046], zoom_start=5)
```

In this line of code we use folium to plot the real map with start location on Iran and zoom=5 in the beginning. You can see the result in underlying figure.



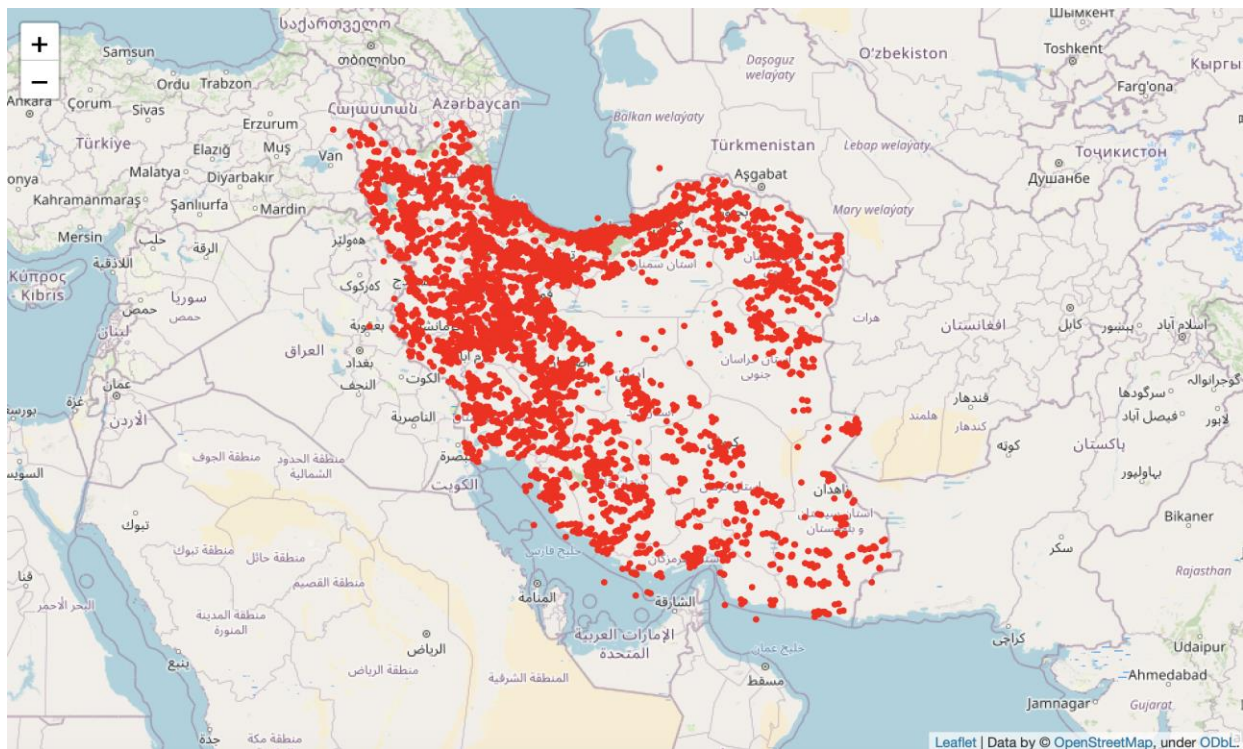
A)

after that we will plot data points which indicate covid-19 patients on this map using a loop on data and `add_to(m)` which adds each point to the real map.

for point in covid_data:

```
folium.Circle(location=point, radius = 1, color='red', fill=True).add_to(m)
```

the figure below is the result of this plot.



we define a method which will do clustering and also plotting data points and clusters with different color on the map:

`-dbscan_on_covid_data(data, minpts, eps) :`

first we do DBSCAN predefined in scikit-learn library on data points and the `fit_predict` method of this object will return the list of labels for all of data points:

-in this list -1 indicates that data is noise or outlier.

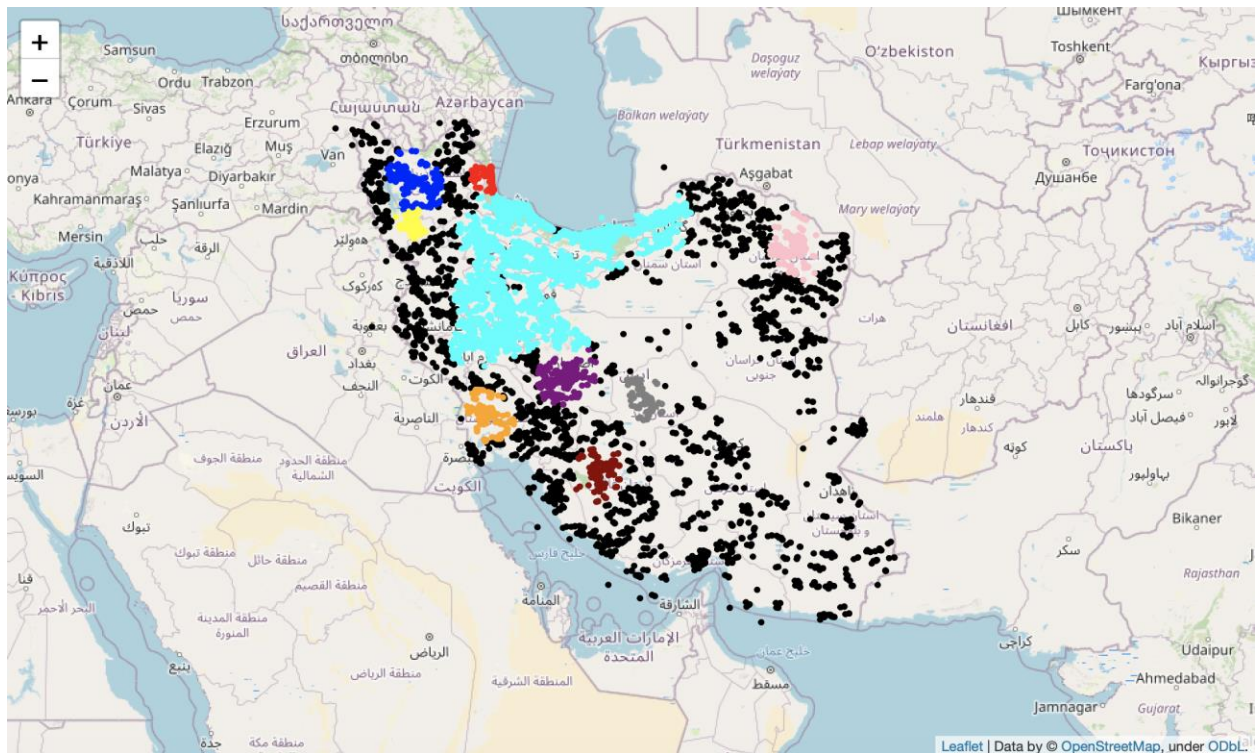
-Other numbers are the cluster number of the data, and data points with same label number belong to same cluster.

B, C, D)

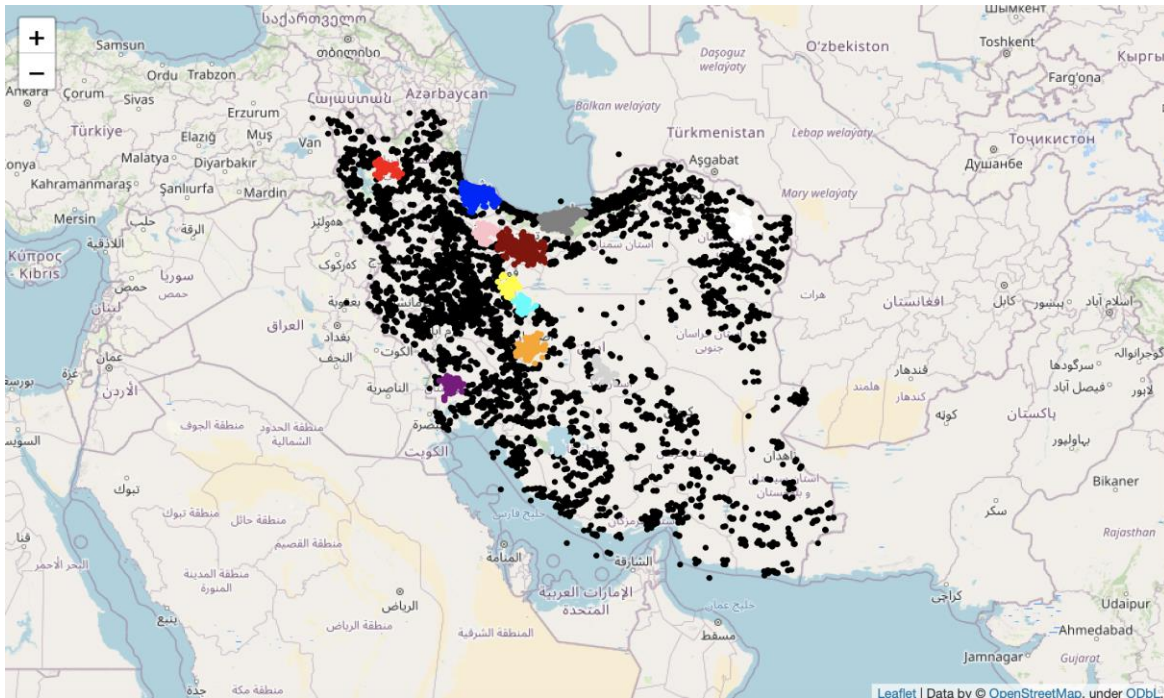
Then we make a map with folium library and add these data points with different colors for each cluster and also the same BLACK color for noise data to the map.

And the method will return the map included clusters and noise data. We choose some arbitrary amounts for “eps” and “minpts” to find the suitable amount for detecting dense clusters which indicates enormous number of patients infected by covid-19 in a specific area on the map.

```
-covid_map = dbscan_on_covid_data(covid_data, minpts=200, eps=0.5)
```

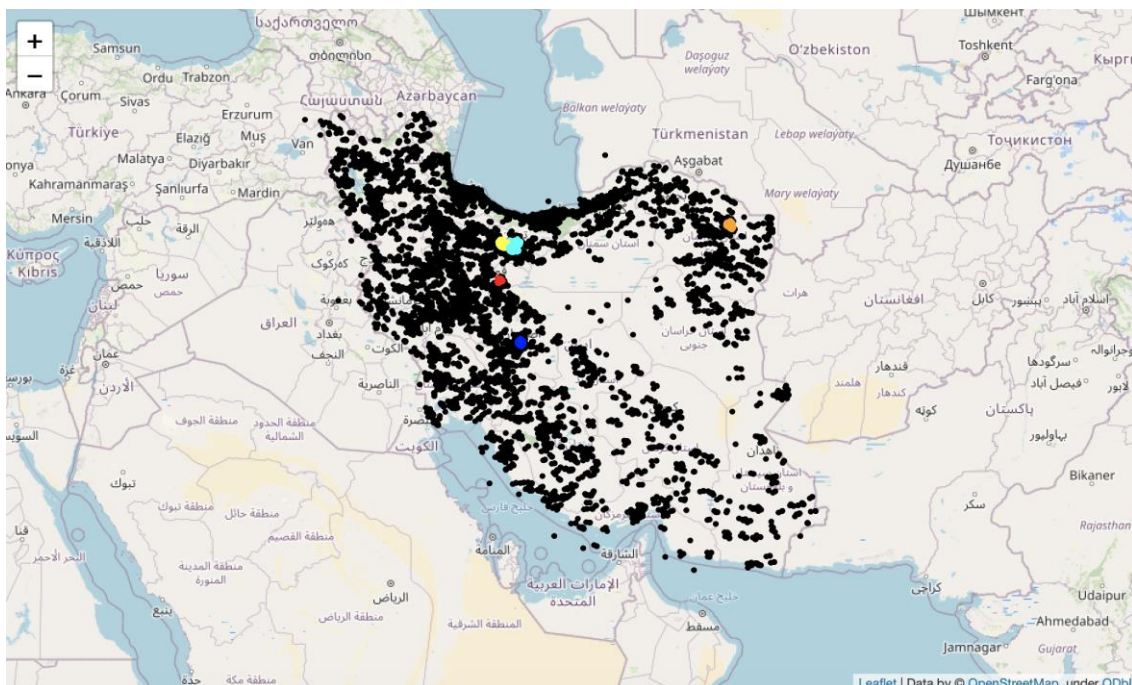


-covid_map = dbscan_on_covid_data(covid_data, minpts=200, eps=0.5)

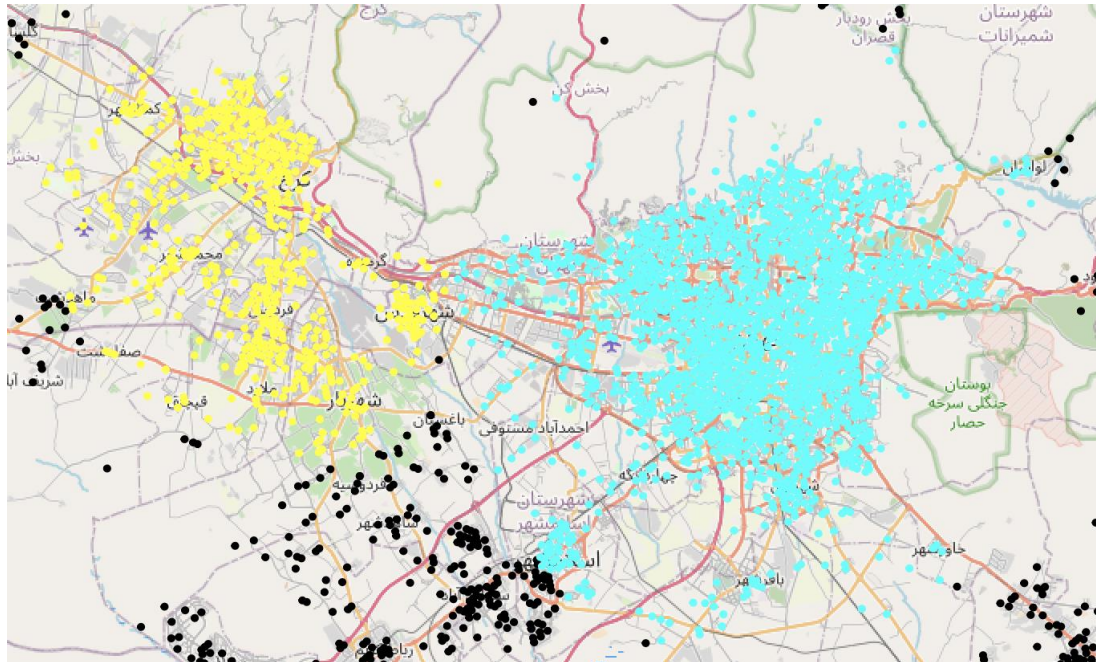


And the proper result that only shows 5 most affected cities by covid-19 :

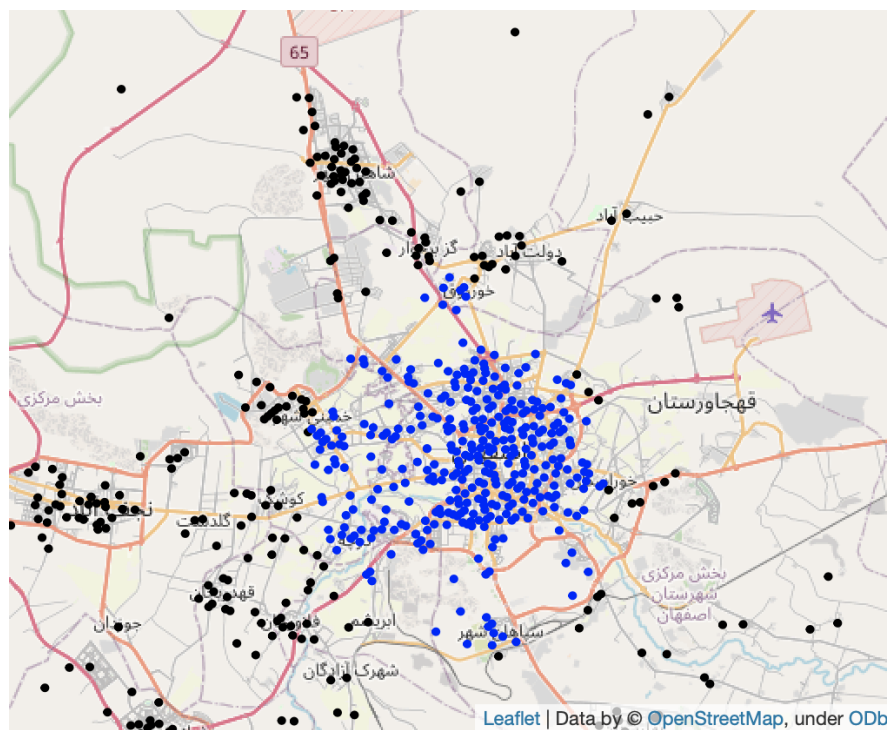
-covid_map = dbscan_on_covid_data(covid_data, minpts=300, eps=0.1)



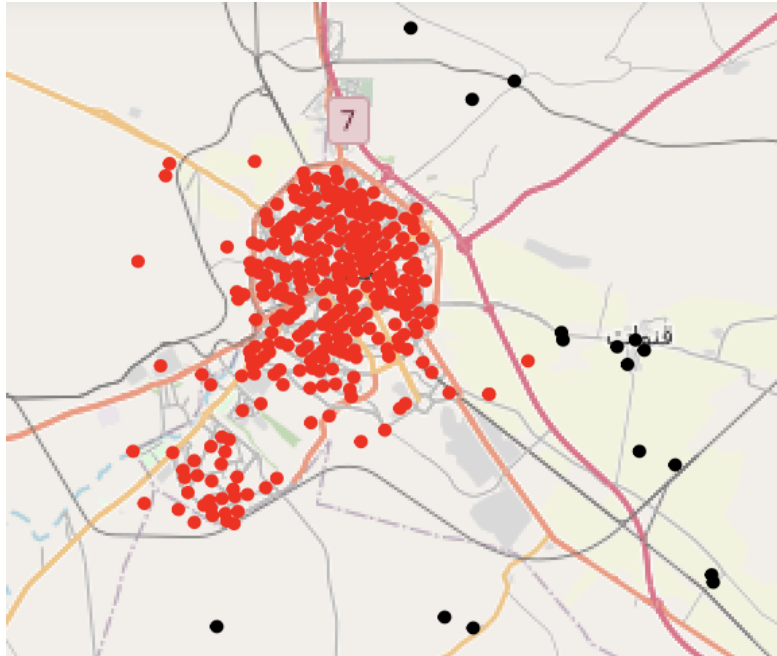
Tehran and karaj:



Isfihan:



Qom:



and Mashhad:

