

# ग्रीष्म/औद्योगिक परियोजना प्रशिक्षण प्रतिवेदन

## Summer/Industrial Project Training Report

on

### Covid-19 Hotspot Detection Using DBSCAN Clustering Method



1 जुलाई 2022 – 31 अगस्त 2022

1<sup>st</sup> July 2022 – 31<sup>st</sup> August 2022

(दो माह/Two Months)

द्वारा प्रस्तुत/Submitted by –

नाम/Name : Prabhakar Jha  
पंजी क्रमांक/Roll No. : 08414803120  
कॉलेज आईडी/College ID : 08414803120  
पाठ्यक्रम/Course : B.Tech  
विश्वविद्यालय/University : Maharaja Agrasen Institute of Technology

पर्यवेक्षक के अधीन/Under the supervision of –

नाम/Name : Mr. Anand Prakash  
पद /Designation : Scientist 'D'  
कार्यालय/Office : ISSA

**पद्धति अध्ययन एवं विश्लेषण संस्थान/  
Institute for Systems Studies and Analyses  
रक्षा अनुसंधान एवं विकास संगठन/  
Defence R&D Organisation  
रक्षा मंत्रालय/Ministry of Defence  
मेटकाफ भवन, दिल्ली / Metcalfe House, Delhi-110054**

**अभ्यर्थी द्वारा घोषणा/**

**DECLARATION BY THE CANDIDATE**

I hereby declare that the work that is being presented by me in this project/study entitled “Covid-19 Hotspot Detection Using DBSCAN Clustering Method” is an authentic record of my own analysis and theoretical research carried out during the period from 1<sup>st</sup> July 2022 to 31<sup>st</sup> August 2022 under the supervision of Mr. Anand Prakash, Scientist ‘D’. (Institute of Systems Studies and Analyses, Defence R&D Organisation, Ministry of Defence, Metcalfe House, Delhi 110054).

**द्वारा प्रस्तुत/Submitted by –**

नाम/Name	:	Prabhakar Jha
पंजी क्रमांक/Roll No.	:	08414803120
कॉलेज आईडी/College ID	:	08414803120
पाठ्यक्रम/Course	:	B.Tech
विश्वविद्यालय/University	:	Maharaja Agrasen Institute of Technology
दिनांक/Date	:	15 August 2022

## अभिस्वीकृति/ ACKNOWLEDGEMENT

I am grateful to Director, ISSA, and Head of HRD for providing me the opportunity to carry out my project at this esteemed organization. I wish to express my deep gratitude to Mr. Anand Prakash, Scientist 'D', ISSA, DRDO for providing guidance and support so far in the project work. This internship report might never have been completed without the necessary practical knowledge, assistance from many books, articles, and websites. I am also grateful to all employees who answered my all questions regarding my study with a smiling face. They helped me in such a way that helped me to feel comfortable there and thus I have completed my report properly. I believe that this endeavour has prepared me for taking up new challenging opportunities in the future.

द्वारा प्रस्तुत/Submitted by –

नाम/Name : Prabhakar Jha

पंजी क्रमांक/Roll No. : 08414803120

कॉलेज आईडी/College ID : 08414803120

पाठ्यक्रम/Course : B.Tech

विश्वविद्यालय/University : Maharaja Agrasen Institute of Technology

दिनांक/Date : 15 August 2022

## पद्धति अध्ययन एवं विश्लेषण संस्थान के बारे में/

### **About the Institute for System Studies and Analyses (ISSA)**

Institute for System Studies and Analyses (ISSA) is a premier institution involved in systems analysis of Defence Systems. It provides analysis support to the top echelons of the three services, SA to RM, and DRDO HQs for scientific decision-making. It also provides systems analysis support to sister labs and other institutions under the Ministry of Defence. ISSA is primarily devoted to systems analysis and specializes in modeling and simulation in wide range of applications.

ISSA adopts state-of-the-art info-technologies such as Computer Networking, Software Engineering, Distributed Database, Distributed Simulation, Web Technologies, Situational Awareness and Soft-Computing Techniques in development of complex simulation products.

## रक्षा अनुसंधान एवं विकास संगठन के बारे में/

### About the Defence Research and Development Organization

DRDO formed in 1958 from the amalgamation of the then, already functioning Technical Development Establishment (TDEs) of the Indian Army and the Directorate of Technical Development & Production (DTDP) with the Defence Science Organization (DSO). Today, DRDO is having more than 50 labs, engaged in developing Defence Technologies covering various disciplines like aeronautics, armaments, electronics, combat vehicles, engineering systems, instrumentation, missiles, advanced computing and simulation, special materials, naval systems, life sciences, training, information systems and agriculture. Over 5000 scientists and about 25,000 other scientific, technical and supporting personnel back DRDO.

#### **Vision**

Make India prosperous by establishing excellent science and technology base and provide our Defence Services decisive edge by equipping them with internationally competitive systems and solutions.

#### **Mission**

- Design, develop and lead to production state-of-the-art sensors, weapon systems, platforms, and allied equipment for our Defence Services.
- Provide technological solutions to the Defence Services to optimize combat effectiveness and to promote well-being of the troops.
- Develop infrastructure and committed quality work force and build strong technology base.

#### **Core Competence**

Department of Defence Research and Development (R&D) is working for indigenous development of weapons, sensors & platforms required by the three wings of the Armed Forces. To fulfill this mandate, Department of Defence Research and Development (R&D), is closely working with academic institutions, Research and Development (R&D) Centers and production agencies of Science and Technology (S&T) Ministries/Depts. in Public & Civil Sector including Defence Public Sector Undertakings & Ordnance Factories.

## विषयसूची/

## CONTENTS

Chapter 1. Introduction	- 8
Chapter 2. Covid 19	- 9
2.1 History of Covid 19	- 9
2.2 definition	- 10
Chapter 3. Related work	- 10
3.1 Density-Based Clustering	- 11
3.2 Clustering Techniques	- 12
Chapter 4. DBSCAN	- 12
4.1 DBSCAN common terminologies	- 12
4.2 Steps Involved in DBSCAN clustering algorithm	- 14
4.2.1 purpose of clustering	- 14
4.3 Parameters	- 15
4.4 DBSCAN steps	- 15
Chapter 5. Perform DBSCAN in Python	- 16
5.1 Get Dataset	- 16
5.1.1 Importing required libraries	- 16
5.2 Exploratory Data analysis	- 17
5.2.1 Checking the head of the data	- 17
5.2.2 Checking the tail of the data	- 18
5.2.3. Describe Dataset	- 19
5.3 Implementing the DBSCAN modal	- 20
Chapter 6. Visualization of DBSCAN clustering	- 22
Chapter 7. Conclusion	- 26



## Chapter 1

### 1. Introduction

The aim of this study is to apply mathematical methods like **density based clustering** DBSCAN on World Novel Corona Virus (Covid-19) Dataset to detect hotspot.

The **COVID-19 pandemic**, also known as the **coronavirus pandemic**. The Novel virus was first identified from an outbreak in Wuhan, China, in December 2019.

The World Health Organization (WHO) declared a Public Health Emergency of International Concern on 30 January 2020 and a pandemic on 11 March 2020.

As of 2 July 2022, the pandemic had caused **more than 548 million cases** and **6.33 million confirmed deaths**, making it one of the **deadliest in history**.

Most people infected with the virus will experience mild to moderate respiratory illness and recover without requiring special treatment. However, some will become seriously ill and require medical attention. Older people and those with underlying medical conditions like cardiovascular disease, diabetes, chronic respiratory disease, or cancer are more likely to develop serious illness. Anyone can get sick with COVID-19 and become seriously ill or die at any age.

The best way to prevent and slow down transmission is to be well informed about the disease and how the virus spreads. Protect yourself and others from infection by staying at least 1 metre apart from others, wearing a properly fitted mask, and washing your hands or using an alcohol-based rub frequently. Get vaccinated when it's your turn and follow local guidance.

The virus can spread from an infected person's mouth or nose in small liquid particles when they cough, sneeze, speak, sing or breathe. These particles range from larger respiratory droplets to smaller aerosols. It is important to practice respiratory etiquette, for example by coughing into a flexed elbow, and to stay home and self-isolate until you recover if you feel unwell.

Computer science is one of the fields, like other fields, which paid attention to COVID-19. People have discussed COVID-19 in all media outlets including social media.

The main process used in computer science for dealing with COVID-19 is Clustering and its algorithms to refer to and deal with. Computer intelligence and digital analysis is also one of the fields that has taken this into consideration using clustering algorithms.

Clustering is an important feature that is also applied to learning data. The unsupervised clustering is defined as segmenting the data into clusters that contain data of the same characteristics, which mainly means sorting the data to make homogenous groups.

Clustering algorithms are applied in different fields namely, image segmentation, data cleaning and exploratory analysis, information retrieval, web pages grouping, market segmentation and scientific and engineering analysis Clustering can also be used as a preprocessing step to identify pattern classes for subsequent supervised classification .



## Chapter 2

### 2. Covid-19

The novel virus resembles MERS coronavirus and SARS coronavirus and it is given the abbreviated name as COVID-19 to stand for CORONA VIRUS DISEASE IN 2019. World Health Organization (WHO) assigned the label pandemic to this disease, which was caused by the SARS-CoV-2 virus, because it is highly infectious and hence became an issue of concern and debate all over the world.

#### 2.1 History of Covid-19

Wuhan in China, in the last two months of 2019, witnessed a pandemic of pneumonia of no obvious causes and origins. However, it was later identified as caused by a new coronavirus (Panwar et al., 2020). It later spread to many other countries rapidly.

This has caused risks to different countries especially those with poor health systems. It is known that pandemics grow with high speed. However, they cannot grow rapidly forever.

Eventually, the virus will finish either because most people have already been infected/killed or because we will obtain the ability to control it. As the situation was different from one country to another due to not following the same restrictions and regulations or not responding to COVID-19 in similar ways, different conditions were posed throughout the whole world. SARS-CoV-2 spreads when people get in close contact with each other especially those infected with it during family and friend gatherings.

To control such virus, early detection of these gatherings and isolation of infected people is a preliminary must. As a vital prevention measure, very new and modern geospatial tools as important digital tools are used to point out the precise locations where patients with COVID-19 reside. These methods support an up-to-date clustering and help in monitoring the spread of COVID-19 in terms of time and space. This can help in creating strategies that can give a good knowledge to epidemiologists and decision makers to intervene on the local levels.

Machine learning, similar to other methods, played an important role in finding out more about the causes and the conditions of the virus. That was an attempt to clean the noisy data spread worldwide in order to inform biological fields where researchers were making every effort to know how the virus lives outside the human body, and to know impact of the different variables like the temperature, populations, and on the spread of COVID-19.

In addition to that the cleaning data led to results that can be useful to control the spread of the virus and help health units to make the right decisions to fight this virus.

## 2.2 Definition

The clustering and classification problems are essential and admired topics of research in the area of pattern recognition and data mining.

The conventional binary and multiclass classifiers are surely not suitable for this target-task mining task because, in this unsupervised learning mode, it is always possible that some of the clusters may not be assigned with any target-class, whereas an increase in the number of target-tasks to solve this problem leads to generation of duplicate information.

These conventional classifiers work fine in the presence of at least two well-defined classes but may become biased, if the dataset suffers from data irregularity problems (imbalanced classes, small disjunct, skewed class distribution, missing values, etc.). Specially, when a class is ill-defined, the classifier may give biased outcome.

## Chapter 3

### 3 Related work

Clustering analysis or simply Clustering is basically an Unsupervised learning method that divides the data points into a number of specific batches or groups, such that the data points in the same groups have similar properties and data points in different groups have different properties in some sense.

Fundamentally, all clustering methods use the same approach i.e. first we calculate similarities and then we use it to cluster the data points into groups or batches. Here we will focus on **Density-based spatial clustering of applications with noise** (DBSCAN) clustering method.

Clusters are dense regions in the data space, separated by regions of the lower density of points. The **DBSCAN algorithm** is based on this intuitive notion of “clusters” and “noise”. The key idea is that for each point of a cluster, the neighbourhood of a given radius has to contain at least a minimum number of point

#### Why DBSCAN?

Partitioning methods (K-means, PAM clustering) and hierarchical clustering work for finding spherical-shaped clusters or convex clusters. In other words, they are suitable only for compact and well-separated clusters. Moreover, they are also severely affected by the presence of noise and outliers in the data.

Real life data may contain irregularities, like:

1. Clusters can be of arbitrary shape such as those shown in the figure below.
2. Data may contain noise

## 3.1 Density-Based Clustering

Partition-based and hierarchical clustering techniques are highly efficient with normal shaped clusters. However, when it comes to arbitrary shaped clusters or detecting outliers, density-based techniques are more efficient.

### Why do we need DBSCAN Clustering?

This is a pertinent question. We already have basic clustering algorithms, so why should you spend your time and energy learning about yet another clustering method? It's a fair question so let me answer that before I talk about what DBSCAN clustering is.

First, let's clear up the role of clustering.

Clustering is an **unsupervised learning** technique where we try to group the data points based on specific characteristics. There are various clustering algorithms with **K-Means** and **Hierarchical** being the most used ones. Some of the use cases of clustering algorithms include:

- Document Clustering
- Recommendation Engine
- Image Segmentation
- Market Segmentation
- Search Result Grouping
- and Anomaly Detection.

All these problems use the concept of clustering to reach their end goal. Therefore, it is crucial to understand the concept of clustering. But here's the issue with these two clustering algorithms.

K-Means and Hierarchical Clustering both fail in creating clusters of arbitrary shapes. They are not able to form clusters based on varying densities. That's why we need DBSCAN clustering.

## 3.2 Clustering Techniques

Clustering is an unsupervised learning approach used iteratively to create groups of relatively similar samples from the population. In this research, Density-based spatial clustering of applications with noise (DBSCAN) clustering techniques have been used to create the clusters of the available samples (research articles):

## Chapter 4

### 4 DBSCAN

Density Based Spatial Clustering of Applications with Noise (abbreviated as DBSCAN) is a **density-based unsupervised clustering** algorithm. In DBSCAN, clusters are formed from dense regions and separated by regions of no or low densities.

DBSCAN computes nearest neighbor graphs and creates **arbitrary-shaped clusters** in datasets (which may contain [noise or outliers](#)) as opposed to [k-means clustering](#), which typically generates spherical-shaped clusters.

Unlike k-means clustering, DBSCAN does not require specifying the number of clusters initially. However, DBSCAN requires two parameters viz. the radius of neighborhoods for a given data point  $p$  ( $\epsilon$  or  $\epsilon$ ) and the minimum number of data points in a given  $\epsilon$ -neighborhood to form clusters (minPts).

DBSCAN is also useful for clustering non-linear datasets.

#### 4.1 DBSCAN common terminologies

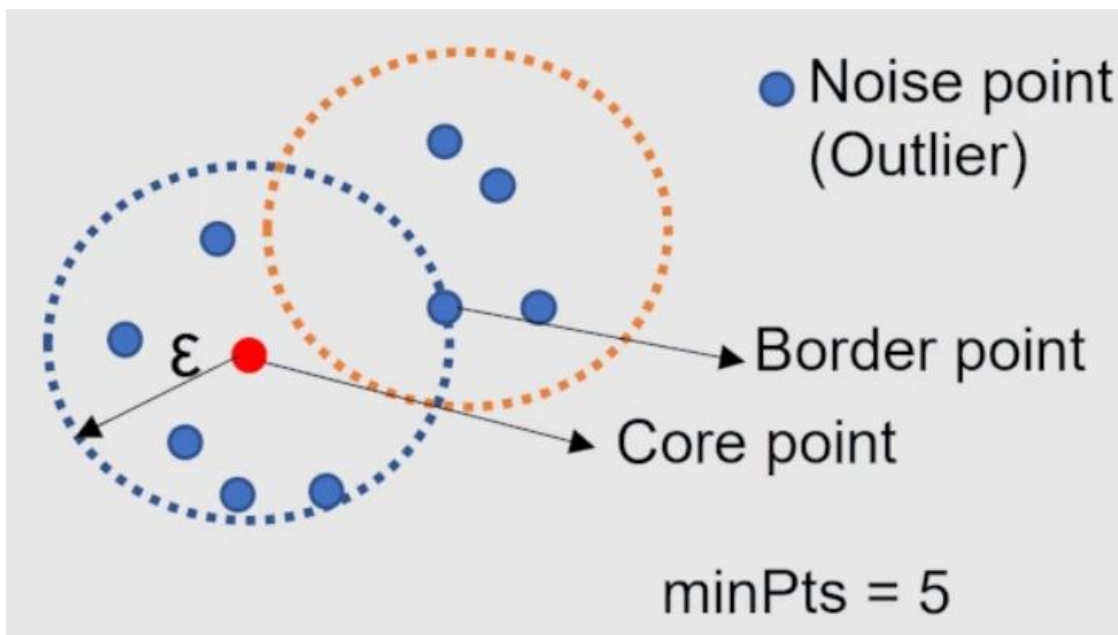
Here we will discuss the core point, border point, noise point (Outlier), and density reachable terminologies used in DBSCAN.

The randomly selected data point  $p$  is called a **core point** if there are more than a minimum number of points (minPts) within a  $\epsilon$ -neighborhood of  $p$ .

#### Types of data points in a DBSCAN clustering

After the DBSCAN clustering is complete, we end up with three types of data points as follows:

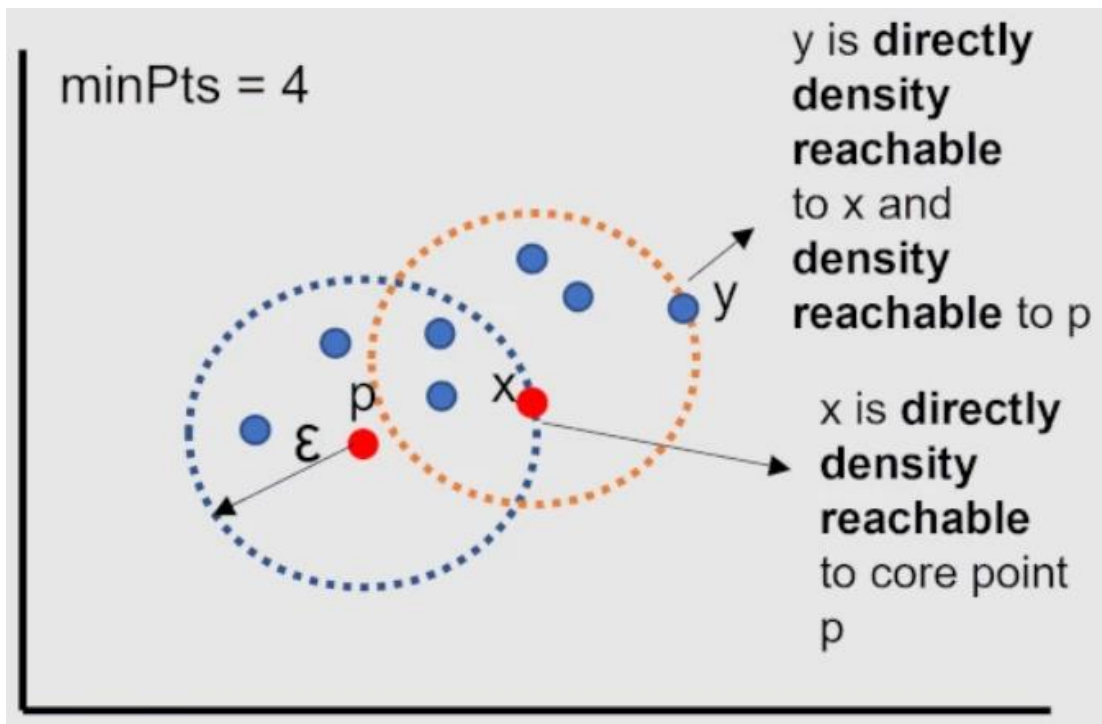
1. **Core:** This is a point from which the two parameters above are fully defined, i.e., a point with at least *Minpoints* within the *Eps* distance from itself.
2. **Border:** This is any data point that is not a core point, but it has at least one *Core point* within *Eps* distance from itself.
3. **Noise:** This is a point with less than *Minpoints* within distance *Eps* from itself. Thus, it's not a *Core* or a *Border*.



A data point is called a **border point** if it is within a  $\epsilon$ -neighborhood of  $p$  and it has fewer than the minimum number of points ( $\text{minPts}$ ) within its  $\epsilon$ -neighborhood.

If a point that is not a core point or border point is called a **Noise point (Outlier)**.

A point  $x$  is **directly density reachable** from point  $p$  if a point  $p$  is a core point and  $x$  is in  $p$ 's  $\epsilon$ -neighborhood. A point  $y$  is **density reachable** from point  $p$  if a point  $y$  is **directly density reachable** to core point  $x$ , which is also **density reachable** to core point  $p$ .



All points within a  $\epsilon$ -neighborhood of a core point belongs to same clusters and all points in a cluster are directly density reachable to core point.

The points which are not density reachable from any core points are called noise (outliers) points.

## 4.2 Steps involved in DBSCAN clustering algorithm

1. Choose any point  $p$  randomly
2. Identify all **density reachable** points from  $p$  with  $\epsilon$  and minPts parameter
3. If  $p$  is a core point, create a cluster (with  $\epsilon$  and minPts)
4. If  $p$  is a border point, visit the next point in a dataset
5. Continue the algorithm until all points are visited

### 4.2.1 purpose of clustering

The **goal of cluster** analysis or **clustering** is to group a collection of objects in such a way that objects in the same group (called a **cluster**) are more like each other (in some sense) than objects in other groups (**clusters**)

In many areas, cluster analysis is used, including pattern recognition, image analysis, retrieval of information, bioinformatics, compression of data, computer graphics, and machine learning.

### 4.3 Parameters

DBSCAN algorithm works with two parameters.

These parameters are:

- 1) **Epsilon (Eps):** This is the least distance required for two points to be termed as a neighbour. This distance is known as Epsilon (Eps). Thus we consider *Eps* as a threshold for considering two points as neighbours, i.e., if the distance between two points is utmost *Eps*, then we consider the two points to be neighbours.
- 2) **MinPoints:** This refers to the minimum number of points needed to construct a cluster. We consider MinPoints as a threshold for considering a cluster as a cluster. A cluster is only recognized if the number of points is greater than or equal to the *MinPts*.

### 4.4 DBSCAN Steps

The following are the DBSCAN clustering algorithmic steps:

- **Step 1:** Initially, the algorithms start by selecting a point (x) randomly from the data set and finding all the neighbour points within *Eps* from it. If the number of *Eps-neighbours* is greater than or equal to **MinPoints**, we consider x a core point. Then, with its *Eps-neighbours*, x forms the first cluster.

After creating the first cluster, we examine all its member points and find their respective *Eps – neighbours*. If a member has at least *MinPoints Eps-neighbours*, we expand the initial cluster by adding those *Eps-neighbours* to the cluster. This continues until there are no more points to add to this cluster.

- **Step 2:** For any other core point not assigned to cluster, create a new cluster.

- **Step 3:** To the core point cluster, find and assign all points that are recursively connected to it.
- **Step 4:** Iterate through all unattended points in the dataset and assign them to the nearest cluster at *Eps* distance from themselves. If a point does not fit any available clusters, locate it as a noise point.

## Chapter 5

### 5. Perform DBSCAN in Python

To perform DBSCAN clustering in Python, you will require to install sklearn, pandas, and matplotlib Python packages.

#### 5.1 Get Dataset

For clustering using DBSCAN, I am using Novel Covid-19 Dataset.

##### 5.1.1 Importing required libraries

Let us begin by importing the required libraries for implementation on the algorithm.

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.cluster import DBSCAN
from collections import Counter
from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
df = pd.read_csv('covidatanew.csv')
```



## 5.2 Exploratory data analysis

This is the process of investigating the available data and determining inconsistencies in patterns and other anomalies with the help of graphical representations and statistical summaries.

### 5.2.1. Checking the head of the data.

```
df.head()
```

- Output

	ID	age	sex	city	province	country	latitude	longitude	geo_resolution	date_confirmation	symptoms	travel_history_location
0	000-1-1	52	male	Shek Lei	Hong Kong	China	22.365019	114.133808	point	14.02.2020	fever	China
1	000-1-10	78	male	Vo Euganeo	Veneto	Italy	45.297748	11.658382	point	21.02.2020	cough	China
2	000-1-100	61	female	Vo Euganeo	Veneto	Singapore	1.353460	103.815100	admin0	14.02.2020	headache	China
3	000-1-1000	61	female	Zhengzhou City	Henan	China	34.629310	113.468000	admin2	26.01.2020	fever,cough	China
4	000-1-10000	61	female	Pingxiang City	Jiangxi	China	27.513560	113.902900	admin2	14.02.2020	fever,cough	China


## 5.2.2. Checking the tail of Data

```
df.tail()
```

### Output:

	ID	age	sex	city	province	country	latitude	longitude	geo_resolution	date_confirmation	symptoms	travel_history_location
7570	007-170856	62	female	Dvorce	Vysocina Region	Czech Republic	49.373590	15.489300	point	21.05.2020	respiratory difficult breathing died same day	Belgium
7571	007-170857	29	male	Rakovnik	Central Bohemian Region	Czech Republic	50.096618	13.720700	point	23.05.2020	respiratory difficult breathing died same day	Qatar
7572	007-172605	15-34	female	Rendsburg-Eckernförde	Schleswig-Holstein	Germany	54.289713	9.781941	point	28.05.2020	respiratory difficult breathing died same day	Qatar
7573	007-174571	15-34	male	Börde	Sachsen-Anhalt	Germany	52.220718	11.347230	point	27.05.2020	respiratory difficult breathing died same day	Qatar
7574	007-21992	0-4	male	Mulheim	Nordrhein-Westfalen	Germany	50.961580	7.005560	point	22.04.2020	respiratory difficult breathing died same day	Qatar

,



### 5.2.3. Describe dataset.

```
df.describe()
```

	latitude	longitude
<b>count</b>	7575.000000	7575.000000
<b>mean</b>	29.991079	19.842965
<b>std</b>	19.705375	76.791685
<b>min</b>	-54.000000	-159.727597
<b>25%</b>	19.187825	-62.086893
<b>50%</b>	32.473749	11.776639
<b>75%</b>	45.745150	85.279570
<b>max</b>	70.071800	174.740000

Next, we check if the dataset has any missing values.

```
# checking for NULL data in the dataset
data.isnull().any().any()
```

## Output

```
False
```

The above output means there are no missing values in our dataset. Since our data is ready to use, let us extract the longitude and latitude columns and apply our DBSCAN model to them.

### 5.3. Implementing the DBSCAN model

As We know, In Dataset 7 and 6 are the column number of longitude and latitude.

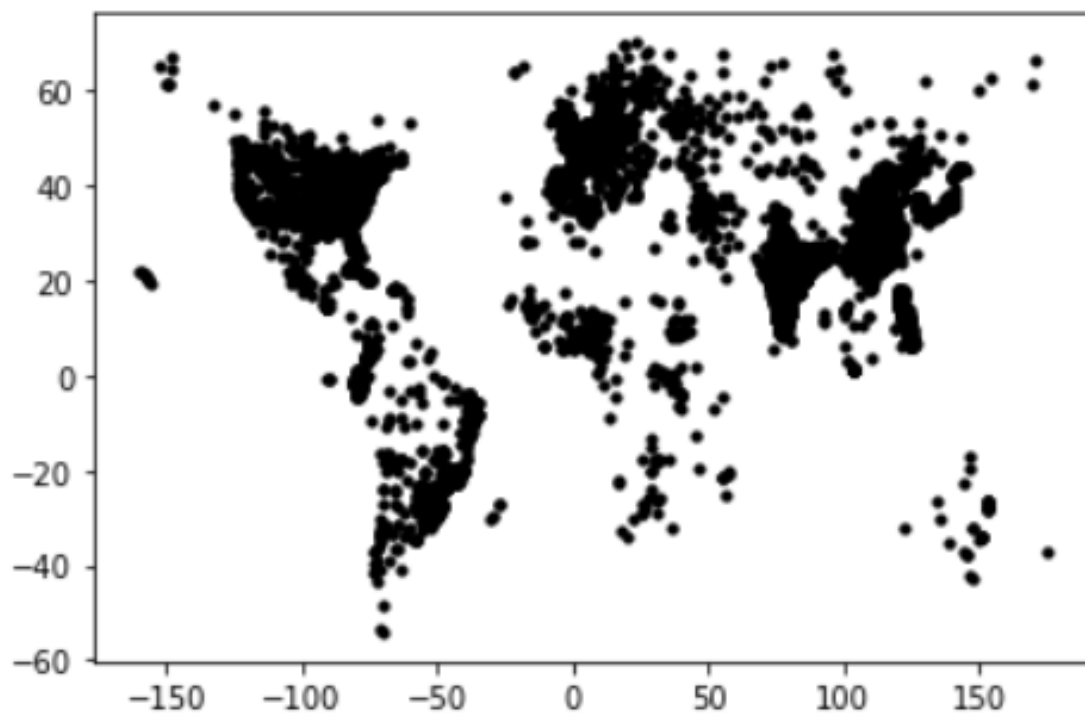
```
df = df.iloc[:, [7,6]].values
```

```
df
```

```
array([[114.133808 , 22.3650193 ],
       [ 11.6583815 , 45.2977477 ],
       [103.8151    ,  1.35346   ],
       ...,
       [ 9.78194123, 54.28971291],
       [ 11.34723024, 52.2207185 ],
       [ 7.00556    , 50.96158   ]])
```

```
plt.scatter(df[:,0], df[:,1], s=10, c="black")
```

```
<matplotlib.collections.PathCollection at 0x2952f1df8b0>
```



```
dbscan = DBSCAN(eps=5, min_samples=5)
```

```
# check unique clusters
```

```
labels = dbscan.fit_predict(df)
```

```
np.unique(labels)
```

```
array([-1,  0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,  
       16, 17, 18, 19, 20, 21, 22, 23, 24], dtype=int64)
```

The most amazing thing about DBSCAN is that it separates noise from the dataset pretty well. Here, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24 are the twenty five different clusters, and -1 is the noise.

## Chapter 6

### 6. Visualization of DBSCAN clustering

Visualize the cluster as a scatter plot and color the clusters using predicted class labels,

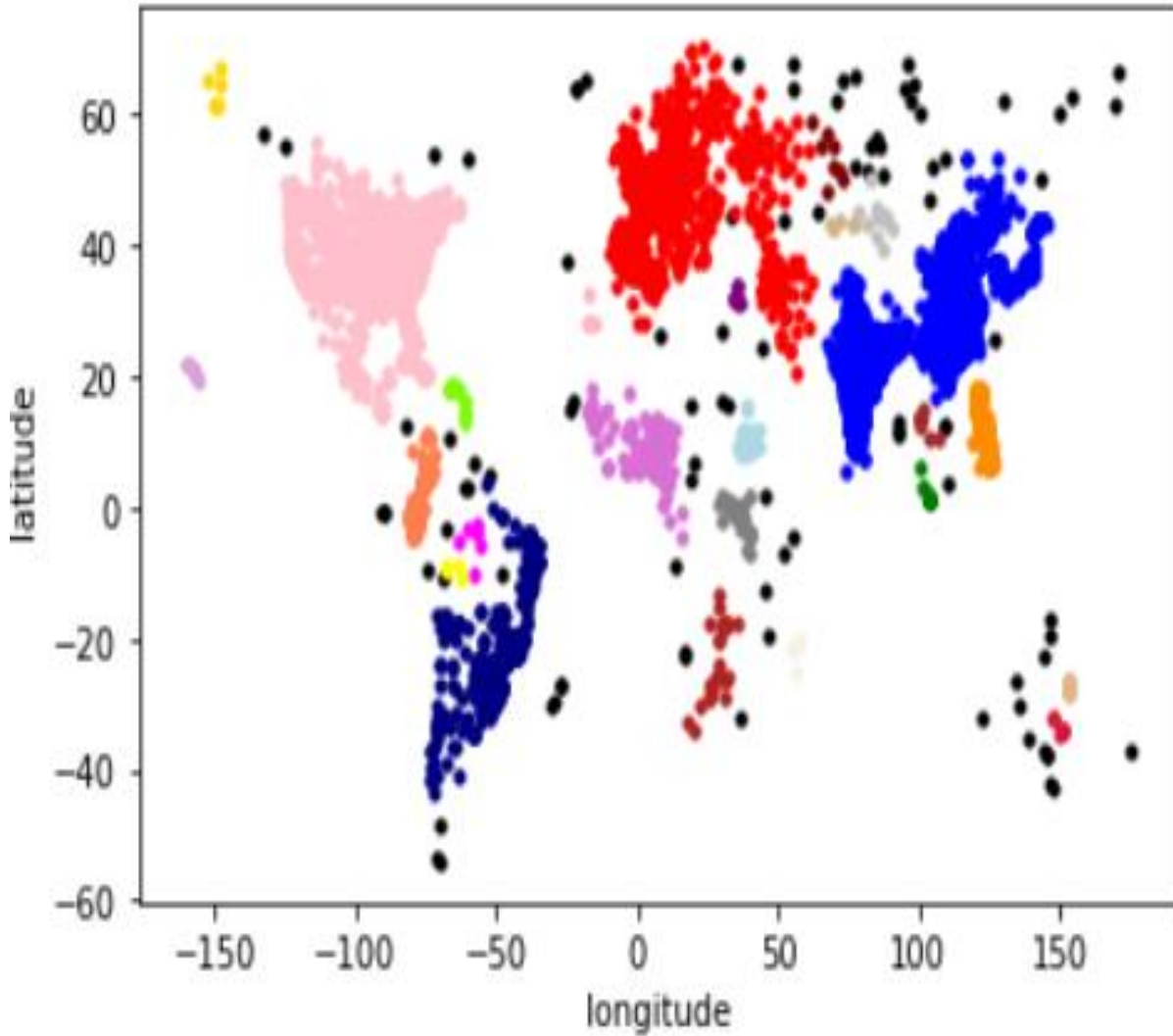
Let's plot the results and see what we get.

```

plt.scatter(df[labels== -1,0], df[labels== -1,1], s=10, c= 'black')
plt.scatter(df[labels== 0,0], df[labels== 0,1], s=10, c= 'blue')
plt.scatter(df[labels== 1,0], df[labels== 1,1], s=10, c= 'red')
plt.scatter(df[labels== 2,0], df[labels== 2,1], s=10, c= 'green')
plt.scatter(df[labels== 3,0], df[labels== 3,1], s=10, c= 'brown')
plt.scatter(df[labels== 4,0], df[labels== 4,1], s=10, c= 'pink')
plt.scatter(df[labels== 5,0], df[labels== 5,1], s=10, c= 'silver')
plt.scatter(df[labels== 6,0], df[labels== 6,1], s=10, c= 'purple')
plt.scatter(df[labels== 7,0], df[labels== 7,1], s=10, c= 'lightpink')
plt.scatter(df[labels== 8,0], df[labels== 8,1], s=10, c= 'navy')
plt.scatter(df[labels== 9,0], df[labels== 9,1], s=10, c= 'burlywood')
plt.scatter(df[labels== 10,0], df[labels== 10,1], s=10, c= 'orchid')
plt.scatter(df[labels== 11,0], df[labels== 11,1], s=10, c= 'crimson')
plt.scatter(df[labels== 12,0], df[labels== 12,1], s=10, c= 'coral')
plt.scatter(df[labels== 13,0], df[labels== 13,1], s=10, c= 'darkorange')
plt.scatter(df[labels== 14,0], df[labels== 14,1], s=10, c= 'firebrick')
plt.scatter(df[labels== 15,0], df[labels== 15,1], s=10, c= 'plum')
plt.scatter(df[labels== 16,0], df[labels== 16,1], s=10, c= 'gold')
plt.scatter(df[labels== 17,0], df[labels== 17,1], s=10, c= 'gray')
plt.scatter(df[labels== 18,0], df[labels== 18,1], s=10, c= 'lightblue')
plt.scatter(df[labels== 19,0], df[labels== 19,1], s=10, c= 'lawngreen')
plt.scatter(df[labels== 20,0], df[labels== 20,1], s=10, c= 'linen')
plt.scatter(df[labels== 21,0], df[labels== 21,1], s=10, c= 'magenta')
plt.scatter(df[labels== 22,0], df[labels== 22,1], s=10, c= 'maroon')
plt.scatter(df[labels== 23,0], df[labels== 23,1], s=10, c= 'tan')
plt.scatter(df[labels== 24,0], df[labels== 24,1], s=10, c= 'yellow')

plt.xlabel('longitude')
plt.ylabel('latitude')
plt.show()

```



Different colours represent different predicted clusters. Black represents noisy points (-1 cluster).

DBSCAN amazingly clustered the data points into twenty five clusters, and it also detected noise in the dataset represented by the black color.

One thing important to note here is that, though DBSCAN creates clusters based on varying densities, it struggles with clusters of similar densities. Also, as the dimension of data increases, it becomes difficult for DBSCAN to create clusters and it falls prey to the Curse of Dimensionality.



## DBSCAN Limitations

- DBSCAN is computationally expensive (less scalable) and more complicated clustering method as compared to simple [k-means clustering](#)
- DBSCAN is sensitive to input parameters, and it is hard to set accurate input parameters
- DBSCAN depends on a single value of  $\epsilon$  for all clusters, and therefore, clusters with variable densities may not be correctly identified by DBSCAN
- DBSCAN is a time-consuming algorithm for clustering

## Pros and Cons of DBSCAN

### Pros:

- Does not require to specify number of clusters beforehand.
- Performs well with arbitrary shapes clusters.
- DBSCAN is robust to outliers and able to detect the outliers.

### Cons:

- In some cases, determining an appropriate distance of neighbourhood ( $\epsilon$ ) is not easy and it requires domain knowledge.
- If clusters are very different in terms of in-cluster densities, DBSCAN is not well suited to define clusters. The characteristics of clusters are defined by the combination of  $\epsilon$ -minPts parameters. Since we pass in one  $\epsilon$ -minPts combination to the algorithm, it cannot generalize well to clusters with much different densities.

## Chapter 7

### 7. conclusion

In this Report, we have covered the DBSCAN algorithm. First, we looked at its key parameters and how this algorithm clusters the data points. We also learned about data points associated with the DBSCAN algorithm. Later we looked at how we implement this algorithm on Covid-19 Dataset. by using DBSCAN, We have detected Covid-19 hotspot.

For more detail about this **Covid-19 Hotspot Detection Using DBSCAN Clustering Method** Report and Dataset and Codes, I am mentioning my Github repository link below:

→ <https://github.com/prabhakar2001/Covid-19-Hotspot-Detection>

### *References*

1. <https://www.section.io/engineering-education/dbscan-clustering-in-python/>
2. <https://www.reneshbedre.com/blog/dbscan-python.html#get-dataset>
3. <https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works/>
4. Dataset source :-<https://www.kaggle.com/datasets/kunwardeepak/covid19-infected-person-list>
5. <https://www.nature.com/articles/s41598-021-88822-3#Sec11>
6. <https://koreascience.kr/article/JAKO202026964745203.pdf>
7. <https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf>
8. <https://arxiv.org/pdf/2004.11706.pdf>
9. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7392081/>