# CLUSTERING OF COUNTRIES BASED ON NUMBER OF COVID-19 CASES BY USING DBSCAN ALGORITHM

**MIN KHANT HTWAY**

**M.C.Sc.** **SEPTEMBER 2022**

# CLUSTERING OF COUNTRIES BASED ON NUMBER OF COVID-19 CASES BY USING DBSCAN ALGORITHM

**By**

**MIN KHANT HTWAY**
**B.C.Sc.**

**A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Computer Science (M.C.Sc.)**

**University of Computer Studies, Yangon**
**September 2022**

# ACKNOWLEDGEMENTS

First of all, I would like to express my innermost gratitude to **Prof. Dr. Mie Mie Khin**, Rector, the University of Computer Studies, Yangon, for allowing me to develop this research and giving me excellent guidance during the period of my dissertation.

Secondly, I would like to express my deepest gratitude to my supervisor **Dr. Hay Mar Soe Naing**, Lecture, Faculty of Computer Science, the University of Computer Studies, Yangon, for her caring and encouragement, and providing me with excellent ideas and guidance during the time of writing this dissertation.

Thirdly, I would like to express special thanks to **Dr. Si Si Mar Win**, Professor, and **Dr. Tin Zar Taw**, Professor, Faculty of Computer Science, the University of Computer Studies, Yangon, as a Dean of Master's Course, for giving me the valuable guidance and suggestions during the development of this thesis.

In addition, I would like to acknowledge and special thanks to **Daw Mya Thandar Aung,** Laecture**,** Head of the Department of English, the University of Computer Studies, Yangon. I would like to thank her for valuable supports and revising my dissertation from the language point of view.

Finally, I would like to extend my thanks to all my teachers who taught me throughout the master's degree course and my friends for their cooperation. The completion of my dissertation would not have been possible without supporting and nurturing of my family.

# STATEMENT OF ORIGINALITY

I hereby certify that the work embodied in this thesis is the result of original research and it has not been submitted for a higher degree to any other University or Institution.

---------- ----------

Date Min Khant Htway

# ABSTRACT

Nowadays, clustering is the important technique for the analysis of data. There are many clustering algorithms. Among them, Density-Based Spatial Clustering of Application with Noise (DBSCAN) is useful for medical domain. Therefore, the clustering of COVID-19 statistic data is implemented by using DBSCAN method. It is a density-based clustering algorithm, grows regions with sufficiently high point density into clusters and discovers cluster of arbitrary shape and size in medical databases. This system clusters in each country occurs the similar number of COVID-19 cases. Three distance measuring methods namely Euclidean, Manhattan and Minkowski are used to calculate the distance between each country and they evaluate the effectiveness of clustering performance in DBSCAN. The silhouette coefficient is used to measure the goodness of clustering quality. According to the experiments, using DBSCAN with Euclidean distance achieved the superior result. This system is implemented by using Python programming language.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF EQUATIONS

# CHAPTER 1
# INTRODUCTION

Data mining is an innovation utilized in various disciplines to look through critical relationships among variables in large datasets. It is utilized to uncover the covered up or obscure data that is not clear, yet at the same possibly helpful. The important data analysis task of clustering is to classify a set of items into plausible homogenous groupings as cluster. In data mining, clustering is a finding progression that groups a set of data so that the similarity between inter-clusters is minimized and the similarity within intra-clusters is maximized.

A measure of object dissimilarity, which may be calculated for a variety of data types, including interval scaled, binary, nominal, ordinal, and ratio scaled variables, or combinations of these variable types, can be used to assess the quality of clustering. The groups that improved and the individuals that should be grouped with them are not necessarily prior knowledge for clustering. The goal of clustering is to increase the degree of similarity between each cluster and the degree of dissimilarity among clusters. Partitioning, hierarchical, density-based, grid-based, and model-based clustering are the five distinct techniques available.

In data mining techniques, clustering can be used in a variety of industries, including statistical data analysis, pattern recognition, image processing, and other business applications, to identify interesting data distributions and patterns in the underlying data. There is still significant work to be done to produce an algorithm or approach for clustering very big databases and high dimensional data, despite the fact that academics have been working with clustering algorithms for decades and that many algorithms have been developed. The DBSCAN algorithm, a standout among clustering algorithms, performs admirably when clustering spatial data.

As a remarkable representative of Density-based clustering algorithms, DBSCAN (density-based spatial clustering of application with noise) algorithm shows a great execution in clustering especially in medical domain. The density-based clustering algorithm is based on the premise that a cluster in a data space is a contiguous region of high density point and is independent of prior knowledge of the number of clusters. Based on the idea that clusters are dense areas in space that are separated from

less dense areas, DBSCAN data clustering divides high density areas into clusters and finds clusters of arbitrary shape in spatial databases with noise.

According to the DBSCAN algorithm, this system identifies clusters of counties that cause the similar number of COVID-19 cases. By using this proposed system, the user can clearly know which country faces the worst case and which country has the similar case with other country in the world.

## 1.1 Objectives of the Thesis

The main objectives of this thesis are:

- To present the DBSCAN clustering algorithm
- To implement the clustering system on COVID-19 statistic data
- To cluster each country that occurs the similar amount of suffered COVID-19 cases
- To point out the clear clustering results that are group of countries in the world under WHO region

## 1.2 Motivation of the Thesis

Clustering is the important technique for the analysis of data. Therefore, the clustering of coronavirus disease 2019 (COVID-19) statistical data is implemented by using DBSCAN (Density-Based Spatial Clustering of Application with Noise) method. DBSCAN is a density-based clustering algorithm, finds clusters in medical databases that can be of any shape or size by grouping together areas with a sufficient amount of density. DBSCAN algorithm is useful for medical dataset. According to the DBSCAN algorithm, this system identifies clusters of counties that cause the COVID-19 disease. By using this COVID-19 clustering system, the user can clearly know which country faces with the worst case and which country has the similar case with other countries in the world.

## 1.3 The System Overview

This system uses the DBSCAN algorithm for clustering of countries based on the number of COVID-19 cases. In this system, the user firstly imports statistical COVID-19 cases dataset for DBSCAN process. Then, the user inputs the minimum number of points (MinPts) value that is the nearest neighbor value. To calculate the

distance between each object, user can choose the distance measuring methods, namely, Euclidean, Manhattan and Minkowski method. According to the user selected distance measure, this system calculates the distance between each object (each country) based on confirmed cases, death cases, recovered cases, active cases, new cases, new death cases and new recovered cases. And then, this system identifies the maximum radius of the neighborhood ($\epsilon$) value based on the resultant k distance graph through the selected distance measure.

By using the given $\epsilon$ and MinPts values, the core point, border point and noise objects are identified. According to the DBSCAN process, this system clusters each core object until all objects have been processed. If an object is a core object, the DBSCAN determines all points expect this point is density-reachable from it and forms a cluster. After clustering each core object, this system identifies noise object that is not included in some clusters. After that, the Silhouette score is calculated to identify the clustering quality of each country. The cluster quality results are different based on the selected similarities calculation methods. This system compares each cluster quality of each country using silhouette scores. Finally, this system displays each cluster, noises and cluster quality results.

## 1.4 Organization of the Thesis

This thesis is organized as five Chapters, abstract, acknowledgement and references. They are as follows:

In **Chapter 1**, Clustering of Countries based on Number of COVID-19 Cases using DBSCAN Algorithms is introduced. This chapter also described objectives of the thesis, motivation, system overview and organization of the thesis.

In **Chapter 2,** the fundamental of Data Mining and Clustering techniques are presented.

In **Chapter 3**, data Normalization, Distance measuring methods, Density based Clustering, DBSCAN are discussed in detail.

In **Chapter 4**, the system design, implementations and DBSCAN method calculations are expressed.

In **Chapter 5**, the conclusion of the thesis work is presented. In addition, advantages and further extensions of the system are depicted.

# CHAPTER 2

# BACKGROUND THEORY

## 2.1 Data Mining

Data mining has garnered considerable interest in the information industry for recent years due to the abundance of huge data and the pressing need to convert this data into useful knowledge and information. Numerous applications for the information and knowledge gathered include business management, production control, market analysis, engineering design, and academic research. Data mining can be considered as a natural progression in the development of information technology.

Data mining is also known as database knowledge discovery (KDD). Generally speaking, it is the process of identifying insightful patterns and information from information sources like databases, text, photos, and the web. The patterns ought to be true, possibly helpful, and comprehensible. Data mining is a multidisciplinary field that includes machine learning, information retrieval, artificial intelligence, statistics, databases and visualization. Data mining applications typically start with a data analyst (data miner) who understanding the application domain. Data analysts identify appropriate data sources and target data.

Tasks involving data mining are numerous. Among the more popular ones are association rule mining, sequential pattern mining, supervised learning (also known as classification), and unsupervised learning (also known as clustering).

- **Supervised Learning**: Both online mining and real-world data mining use supervised learning the most frequently. It also goes by the name of classification. By using data that has been labeled with predetermined classes or categories, it seeks to train a classification function. Future occurrences of data are classified into these classes using the resultant classifier. The process is known as supervised learning because the data examples used for learning, or training data, are labeled with preset classes.
- **Unsupervised Learning**: Unsupervised learning uses data without established classifications for learning. The learning algorithms need to uncover any hidden patterns or structures in the data. Clustering, which groups or clusters data

instances based on their shared or unique qualities, is one of the popular unsupervised learning approaches.

- **Association Rule Mining**: An essential group of data regularities is the category of association rules. An elementary data mining endeavor is the exploration of association rules. The database and data mining communities have likely researched it the most and consider it to be the most significant model ever created. Finding all co-occurrences, also known as associations, between data elements is its goal. Market basket data analysis, which seeks to understand how the items consumers purchase in a supermarket are related, is a typical example of an association rule mining application.
- **Sequential Pattern Mining**: The sequence of transactions is ignored by association rule mining. Such orderings are important in many applications, though. Understanding whether individuals purchase certain things in a particular order is important for market basket analysis. For instance, purchase the bed before the linens. These apps will not be suitable for association rules. There must be a logical progression. Data sets that commonly appear together in a series are discovered by sequential pattern mining [6].

### 2.1.1 Knowledge Discovery Process form Data (KDD)

Even with the exponential development of data storage, the knowledge or information utilized for commercial decision-making is inadequate. Data mining is sometimes referred to as knowledge. Predicting current and future behavior is made possible by the knowledge that has been retrieved. This makes it possible for business owners to make wise decisions based on their expertise. Numerous industries, including aerospace and education, have used data mining. Statistical, mathematical, and other approaches are used to extract knowledge from historical data.

Data mining method known as Knowledge Discovery Process from Data (KDD) is used to discover patterns in data. Each technique serves a distinct purpose and entirely controls how the KDD process turns out. A knowledge discovery process typically consists of the iterative processes listed below:

- **Data cleaning**: This stage deals with noisy, incorrect, missing, or unimportant data.

- **Data integration**: It is possible to combine several heterogeneous data sources into one.
- **Data selection**: The database is searched for the information necessary for the analysis task.
- **Data transformation**: Summarization or aggregation techniques are used to convert or aggregate data into mining-ready formats.
- **Data mining**: Applying intelligent algorithms to extract patterns from data involves this crucial step.
- **Pattern evaluation**: The purpose of this stage is to find highly intriguing patterns that reflect the knowledge base based on some interest metric.
- **Knowledge presentation**: In knowledge presentations, fascinating patterns are shown using visualization techniques, which also aid viewers in comprehending and interpreting the resulting patterns.

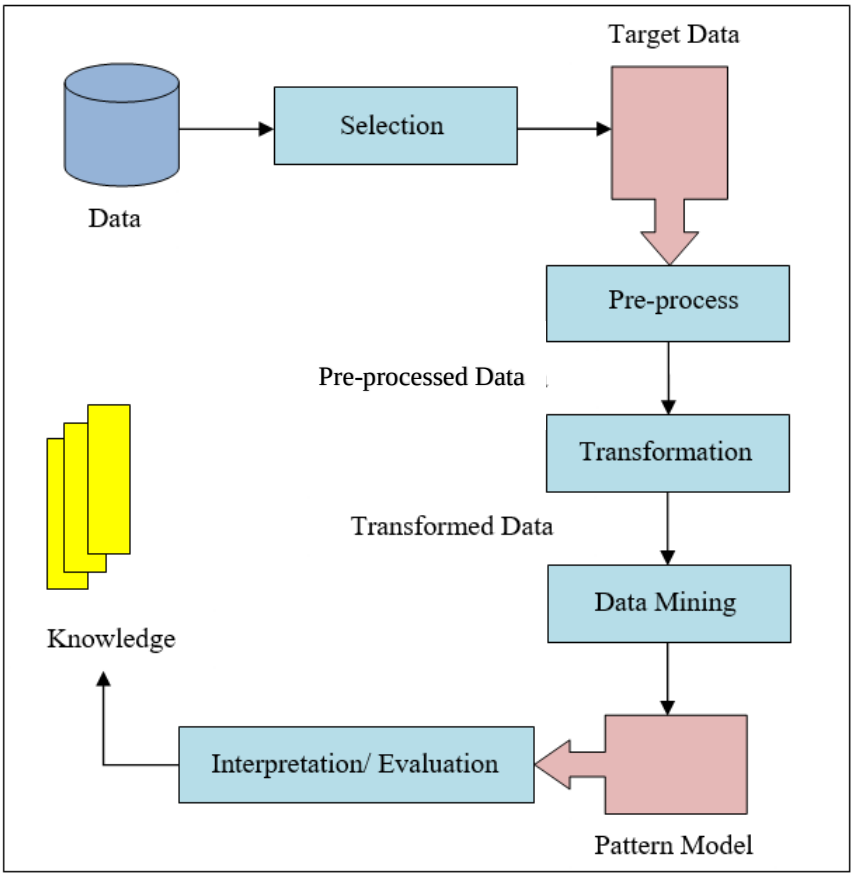


**Figure 2.1 Knowledge Discovery Process from Data (KDD)**

Due to the widespread availability of relational database systems and data warehouses, the first four processes—data cleansing, data integration, data selection,

and data transformation—can be completed by creating a data warehouse and running basic OLAP operations on the created data warehouse. Data mining is a procedure that, on occasion, combines the knowledge representation, pattern assessment, and data mining operations into a single (potentially repetitive) process. Development of systems that utilise information and knowledge sensibly is currently necessary due to the databases' fast expansion. As a result, research into data mining technologies has grown in significance. Depending on personal commercial goals, the significance of this enormous collection of data is extremely debatable [8].

### 2.1.2 Data Mining Application Area

With the development of many organizations across numerous countries, the sorts of data are evolving. Data mining is used in the following industries as a result of this data diversification:

- **Business sector**: To aid in making future company decisions, data mining is utilized to study performance, profitability indicators, and consumer feedback assessments.
- **Marketing and retailing sector:** Retail shop managers may recognize their loyal consumers, issue discounts, and organize shelves in accordance with client needs thanks to data mining's reliable information on customer purchasing habits and top-selling items.
- **Bio-informatics:** In the biomedical area, data mining compiles patient medical information to identify connections between illness and treatment results while assessing genetic and proteomic data.
- **Climatology:** In order to anticipate future meteorological patterns and detect natural risks like cyclones, data mining analyzes historical weather data.
- **Banking and Finance:** Individual banking records are analyzed using data mining to provide various marketing tactics, loan approvals, stock forecasts, and checking for various sorts of fraud and money laundering against target client groups.
- **Security and data integrity:** When a security breach or intrusion of any kind is discovered, data mining may be utilized to monitor various systems and send out alarms. It can assist in identifying the root of firewall security problems.

- **E-Commerce:** In order to assist up-selling and cross-selling, data mining techniques are employed in e-commerce to examine client search habits.
- Forensic and criminal investigation: In forensic and criminal departments, data mining is used to examine prior criminal records in order to identify offenders and ascertain their patterns of criminal behavior and attitudes.
- **Government records:** The analysis of government records using data mining techniques. It is utilized to produce data that is particular to a citizen, including information on their career history, medical histories, law enforcement activities, and analysis of tax fraudsters.
- **Cloud computing**: One of the primary sources of data for all types of applications today may be regarded as cloud computing. Cloud servers also offer speed, dependability, efficiency, and security. Individual infrastructure costs are decreased with these servers. Because of this, KDD approaches and other data mining algorithms can even produce unique search patterns and applications to uncover any information remaining concealed in unstructured data [8].

### 2.1.3 Unsupervised Learning

Learning under supervision identifies patterns in the data that link class qualities to data attributes. The values of the class attribute for upcoming data instances are then predicted using these patterns. These classes are a representation of some actual prediction or classification jobs, such as figuring out if a news item is under the political or sports category or whether a patient has a specific illness.

Nevertheless, the data in certain apps lacks class properties. The user wants to investigate the data to discover any innate structure. A method for locating this structure is clustering. Data instances are grouped together in sets called clusters so that they are similar to one another but quite distinct from one another when they are in separate clusters.

Since no class values are provided to indicate previous divisions or groups of data, unlike supervised learning, clustering is frequently referred to as unsupervised learning. One of the most often applied data analysis techniques has proven to be clustering. It also has used a lengthy history for almost every field including marketing,

archaeology, biology, medicine, psychology, botany, sociology, insurance, and library science [6].

## 2.2 Clustering

One of the essential KDD strategies is clustering. It's frequently employed to reveal hidden patterns beneath a group of things. Clustering is a powerful and popular tool for discovering structure in data. It is also used as a tool for understanding and exploring large datasets. Clustering is a popular data analysis activity that aims to arrange a set of items into clusters that are relatively homogenous. Data mining applications frequently utilize clustering to find patterns in the underlying data. In data mining, clustering is a finding technique that arranges a set of data in such a way that intra-cluster and inter-cluster similarity are maximized [6]. Figure 2.2 presents examples of several clustering techniques.

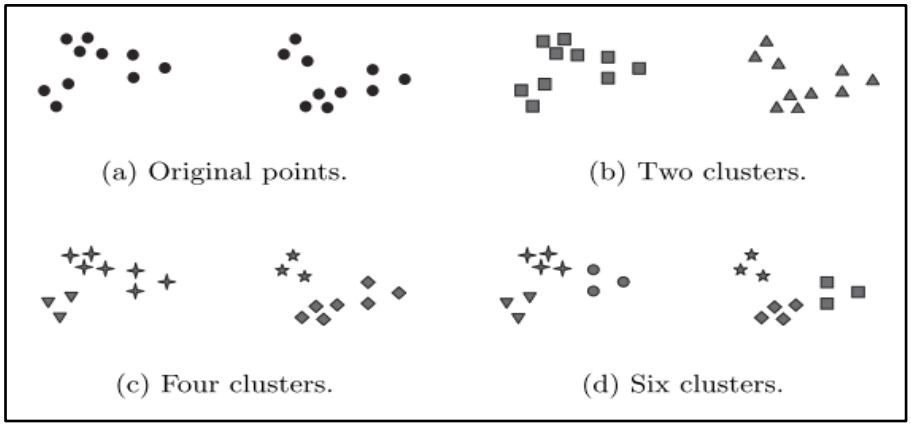


**Figure 2.2 Different Ways of clustering**

## 2.3 Cluster Analysis

Data segmentation, commonly referred to as cluster analysis, has several objectives. They entail classifying or dividing a set of objects (also known as observations, people, cases, or rows of data) into subsets or "clusters" so that the items in each cluster are more closely connected with one another than the objects allocated to separate clusters.

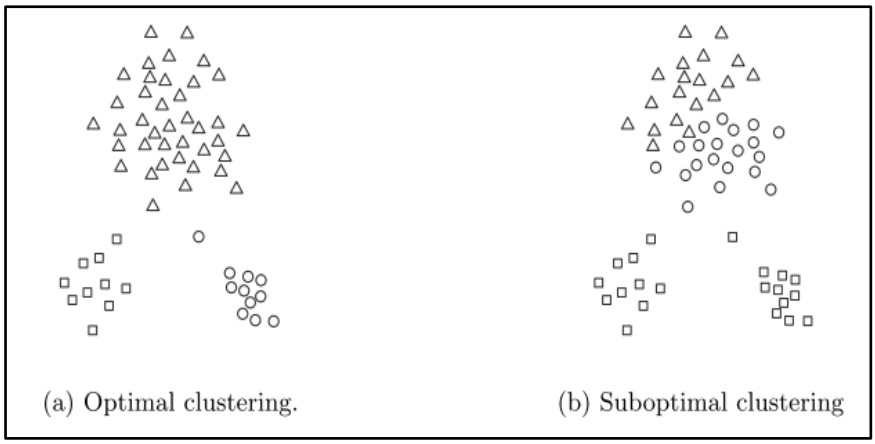


**Figure 2.3 Optimal and Non-optimal Clusters**

The idea of similarity (or dissimilarity), depending on the data and application, between the individual objects being clustered, lies at the core of all cluster analysis objectives. To identify classes (clusters), many similarity measures can be utilized, where the similarity measure influences how the clusters are produced [6]. Both ideal and undesirable clusters are shown in Figure 2.3.

- **Information Retrieval:** There are billions of online pages on the World Wide Web, and a search engine query can yield thousands of pages. These search results can be grouped via clustering into fewer clusters, each of which computes a different feature of the query. For instance, a search for "movies" may produce web pages categorized by topics like trailers, stars, reviews, and theaters. The hierarchy that comes from the division of each category (cluster) into subcategories (sub-clusters) facilitates the user's investigation of the query results [7].

### 2.3.1 Cluster Applications

Market analysis, data analysis, pattern identification, and image processing are just a few of the areas where cluster analysis has been extensively applied. Unsupervised learning includes clustering and its data can divide into groups (clusters) using cluster analysis that are relevant, practical, or both. Clusters will capture the innate structure of the data if meaningful groups are the aim.

However, cluster analysis can occasionally only serve as a helpful starting point for other tasks, including data aggregation. Cluster analysis has long been crucial in a variety of domains, including pattern recognition, machine learning, data mining,

psychology, the social sciences, biology, and statistics, as well as information retrieval. Clusters are prospective classes in the context of analyzing data, and the study of strategies for automatically discovering classes is known as cluster analysis. As examples, consider the following:

- **Biology:** For years, many biologists have been developing taxonomies—hierarchical classifications—of all living things, including kingdom, phylum, class, order, family, genus, and species. Because of this, it may not come as a surprise that most of the early research in cluster analysis focused on developing a branch of mathematics called mathematical taxonomy that could discover these taxonomic structures automatically. Recently, clustering has been used by biologists to examine the vast amounts of genetic data that are now accessible. The discovery of gene clusters with related functions, for instance, has been made possible through clustering.
- **Climate:** It's important to look for trends in the oceans and atmosphere to understand the Earth's climate. Thus, cluster analysis has been used to discover atmospheric pressure patterns in the Polar Regions and areas of the ocean that have a substantial impact on the climate on land.
- **Psychology and Medicine:** Cluster analysis can be used to locate the various subcategories of an illness or condition, which frequently have multiple variants. To distinguish between various forms of depression, for instance, clustering has been utilized. The detection of spatial or temporal patterns in the distribution of disease can also be done using cluster analysis.
- **Business:** Businesses gather a ton of data on their clients, both present and potential. Customers can be divided into smaller groups using clustering to facilitate further analysis and marketing efforts [7].

### 2.3.2 Requirements of Clustering

The study of clustering is a difficult field of research, and all prospective applications come with their own unique set of criteria. Here are some standard criteria for clustering in data mining [6]:

- **Scalability:** With small datasets having fewer than 200 data elements, many clustering techniques perform effectively. Millions of things, however, may

be present in huge databases. A sample of some huge datasets used for clustering can produce skewed results. Clustering methods that are highly scalable are necessary.

- **Ability to deal with different types of attributes:** Numerous algorithms are created to cluster data that is interval-based (numeric). The clustering of different data types, such as binary, categorical, and ordinal data, may be necessary for applications.
- **Discovery of clusters with arbitrary shape:** Many clustering methods base their cluster selections on Manhattan or Euclidean distance metrics. Such distance estimates have the tendency to lead to the discovery of spherical clusters with comparable size and density. However, clusters can take on any shape. It's crucial to create an algorithm that can find clusters of any shape.
- **Minimal domain knowledge to determine input parameters:** In order to perform a cluster analysis, several clustering algorithms demand that the user input specific parameters. The output of clustering can be quite sensitive to the input parameters. It can be challenging to estimate parameters, especially for datasets with high-dimensional objects. This not only burdens the user but also makes it challenging to manage the clustering quality.
- **Ability to deal with noisy data:** The majority of databases in the actual world have outliers or missing, erroneous, or unknown data. Consequently, certain clustering algorithms are susceptible to such data and may produce low-quality grouping.
- **Incremental clustering and insensitivity to the order of input records:** The order of the incoming data can affect some clustering techniques. A similar algorithm might produce distinct clusters for a given set of input objects depending on the order in which those objects are displayed. The creation of algorithms that are indifferent to input order and incremental clustering methods is crucial.

- **High dimensionality:** It is possible for a database or data warehouse to have numerous dimensions or properties. Numerous clustering algorithms excel at working with data that has only two or three dimensions.
- **Constraint-based clustering:** Under various restrictions, a real-world program could need to conduct clustering. It can be difficult to locate data sets that exhibit appropriate clustering behavior and adhere to predetermined restrictions.
- **Interpretability and usability:** Users anticipate that the clustering results will be readable, useful, and clear. It can be necessary to link the clustering to certain semantic applications and interpretations. It is crucial to look into how the application goals may influence the clustering method selection [8].

## 2.4 Clustering Methods

Clustering methods are widely used. The type of data that is available, as well as the precise goal and application, affect the clustering technique that is used. When using cluster analysis as a descriptive or exploratory technique, different algorithms can be tested on the same data to see what the results show. General classifications of the main clustering techniques include partitioning techniques, hierarchical techniques, grid-based techniques, model-based techniques, and density-based techniques [6].

### 2.4.1 Partitioning Method

A partitioning method divides the data into k divisions, each of which represents a cluster and where k<= n. In other words, it divides the data into k groups that, taken collectively, meet the following criteria: (1) There must be at least one item in each group, and (2) every item can only belong to one group. The partitioning method builds the initial partitioning after receiving a specification for the number of partitions to construct. The next step is an iterative rearrangement strategy, which attempts to improve the partitioning by shifting objects from one group to another. Items in the same cluster are typically "near" or connected to one another, whereas objects in other clusters are "far away" or quite dissimilar from one another. This is a common indicator of effective partitioning [6].

### 2.4.2 Hierarchical Method

A given set of data objects is divided into hierarchical levels via hierarchy methods. It can be categorized as either agglomerative or divisive depending on how the hierarchical decomposition is created. Each object forms its own group at the beginning of the aggregation strategy, often known as the bottom-up approach. When groups or objects are close to one another, it combines them until the end condition is satisfied or all groups have been merged into one. Start with all of the objects in the same cluster when using the divisive strategy, also known as the top-down approach. Clusters are divided into smaller clusters on each subsequent iteration until all objects are in a cluster or the termination condition is satisfied [6].

### 2.4.3 Grid-based Method

Grid-based approaches divide the object space into an infinite number of grid-like cells. In a grid structure, all clustering actions are carried out. This method's key benefit is its quick processing time, which is typically independent of the quantity of data items and solely reliant on the quantity of cells per dimension in the quantization space [6].

### 2.4.4 Model-based Method

Model-based approaches look for the best data fit to a certain model by assuming a model for each cluster. A density function reflecting the spatial distribution of data points must be built by a model-based approach to locate clusters. Furthermore, it results in a technique for automatically calculating the number of clusters using common statistics while taking into consideration "noise" or outliers, and these techniques produce reliable clustering techniques [6].

### 2.4.5 Density-based Method

The majority of partitioning techniques group things according to their proximity. Only spherical clusters can be found using these approaches, and finding clusters of any shape is challenging. On the basis of the idea of density, other clustering techniques have been created. They generally believe that a cluster should continue to expand as long as its density—the quantity of objects or data points—exceeds a certain limit. There must be a minimum quantity of points within a specified radius of each

data point in a specific cluster. This technique can be applied to find clusters of any shape and filter out noise (outliers). A common density-based technique that creates clusters based on a density threshold is called Density-Based Spatial Clustering of Applications with Noise (DBSCAN). Additionally, a density-based technique called Ordering Points to Identify the Clustering Structure (OPTICS) computes enhanced cluster rankings for interactive and automated cluster analysis [6].

## 2.5 Related Works

When employing healthcare datasets, O. Godwin and F. N. Ugwokr [1] described the optimized group clusters. Additionally, based on the Silhouette score values, they compared the K-means and DBSCAN clustering algorithms. They used various cluster sizes (K) and distance metrics to examine the K-means algorithm's performance. The performance of the DBSCAN method was then examined using various distance metrics and the least number of points (minPts) needed to establish a cluster. They claimed that there is significant intra-cluster cohesiveness and inter-cluster separation for both the K-means and DBSCAN algorithms.

Fuzzy C-means clustering, spatial fuzzy C-means clustering, K-means, Gustafson kessel clustering, and DBSCAN were all compared by D. J. Divya and S. Prakasha in 2019 [2]. With the use of multiple clustering techniques and a variety of picture segmentation settings, they were able to identify a tumor in a human brain MRI. They looked at a variety of metrics, including information variation, the Rand index, and global consistency error, to examine the accuracy of tumor identification.

The BIRCH, K-means, agglomerative clustering was compared in 2021 by V. Crnogorac, M. Grbic, and M. Dukanovic [3]. According to the amount of European COVID-19 patients, European countries are clustered. Three alternative clustering algorithms were used to do the clustering, which was based on publicly available data provided on the website of the European Centre for Disease Prevention and Control. Value of the Silhouette Coefficient is used to estimate how well a cluster can cluster data. Officers and practitioners in public health may find the findings valuable.

# CHAPTER 3

# DENSITY BASED CLUSTERING

## 3.1 Density-based Clustering

Similar to partitioning techniques, the density-based clustering algorithms describe a dataset by turning each instance into a point and using the information properties of the source set. As with its partitioning predecessor, the plane has clusters with high inward and low outside densities. Iterative object addition to clusters is the most typical strategy, which is unusual for partitioning techniques. This allows for the examination of each point's closest neighbors, the framing of arbitrary forms, and the convergence of preexisting clusters as the algorithm traverses all of the points. Consequently, analysis has the ability to cluster data object sets that include hollow formations while also effectively distinguishing noise instances from important data [10].

Due to their ability to deal with noise and ability to select clusters of arbitrary shapes, density-based algorithms produce the benefits over other approaches. The following are density-based clustering algorithms:

- **DENCLUE:** DENsity based CLUstEring (DENCLUE) uses influence functions during points to depict the information space and is dependent on a set of density distribution functions. The method has a solid mathematical foundation that helps set representation, but the selection of the density and noise parameters might negatively affect the average linear time complexity.
- **OPTICS:** Similar to DBSCAN, Ordering Points to Identify the Clustering Structure (OPTICS) gives enhanced cluster ordering rather than characterizing actual clusters. This method is quite useful because the time complexity is comparable to DBSCAN; averaging O; and the ordering can be used to both identify important cluster characteristics and analyze the development of the cluster space (n log n).
- **GDBSCAN:** The Generalized Density-Based Spatial Clustering of Applications with Noise (GDBSCAN) technique can use non-spatial qualities rather than the quantity of objects to describe neighborhood density. It can cluster both polygons and points. The process achieves independence from a

local input variable by not limiting clusters to a maximum radius. Time complexity is comparable to DBSCAN, although O(n log n) can be achieved with only RTrees [10].

- **UltraDBScan:** Dynamic cluster radius estimation is a concept introduced by UltraDBScan. Dynamic noise removal is made possible by altering this number in response to the cluster's point count as the algorithm passes through each point. The declaration of valid points may appear as noise in widely scattered sets, but it is incredibly helpful in sets with strongly bound clusters.
- **DBSCAN:** The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm identifies and generates areas with the appropriate density as determined by input parameters. The approach can suffer from poor input parameter selection because it does not need that all points be assigned to clusters, making it resilient to noise and supporting clustering of arbitrary shapes. Normally, time complexity is O (n log n), however without a spatial index, it can be as bad as quadratic. DBSCAN is frequently used because it can manage noise and define clusters with irregular forms. [4].

### 3.2 Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Density-Based Spatial Clustering of Applications with Noise is known as DBSCAN. It is a clustering algorithm based on density. It creates clusters out of regions with a sufficient high density, finds clusters with any shape using density-based methods, and produces clusters out of areas with noise-filled spatial datasets.

The DBSCAN computation rests on the assumption that all data points in a data collection can be divided into two categories: clusters and noise. The neighborhood area of an object (Eps) is defined as a set of densely connected regions with a specific radius that contain at least a specified number of points (MinPts) that must exit in the Eps. Data are viewed as noise if a region has fewer points than a predetermined threshold. The *Eps* and *MinPts* values are needed to specify first. The density of the cluster that must be recovered is then controlled by these threshold values. Figure 3.1 displays a cluster of *Eps* and *MinPts*.

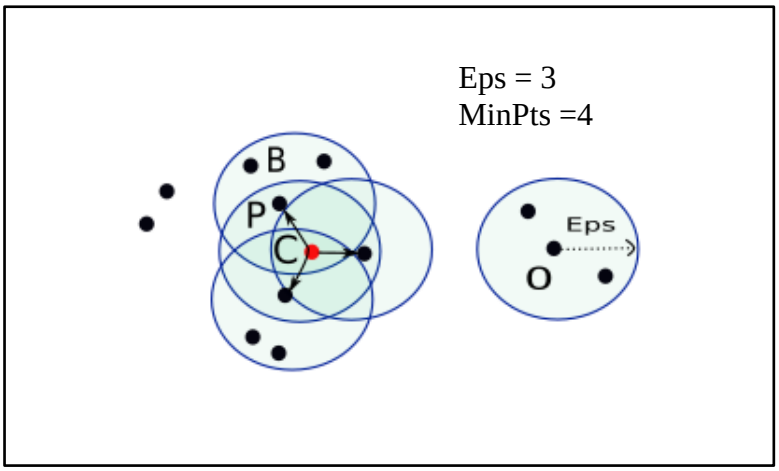


**Figure 3.1 Cluster with *Eps* and *MinPts***

The DBSCAN clustering procedure is based on the distinction between core, border, and noise points in the dataset.

- **Core Point:** A core point is a location within a cluster, i.e., a location with an Eps neighborhood of at least *MinPts*.
- **Border Point:** Border points are points that are located between clusters. Compared to *MinPts*, it has a smaller Eps neighborhood.
- **Noise:** Any point that neither forms the core nor the border is considered as noise. Figure 3.2 displays a core point, boundary, and noise [11].

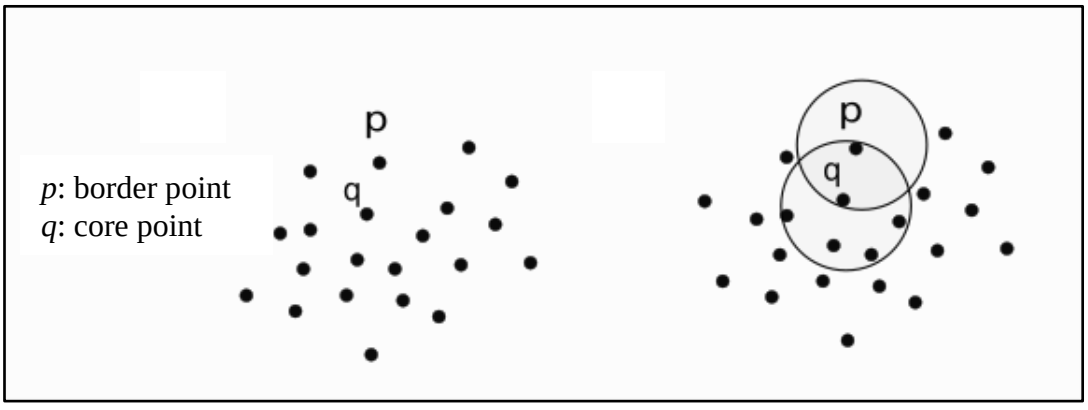


**Figure 3.2 Core point, Border point and Noise**

### 3.2.1 DBSCAN Algorithm

In order to discover clusters and noise in a spatial database, DBSCAN (Density Based Spatial Clustering of Applications with Noise) calculation is used. DBSCAN starts at any point *p* and recovers every point that is density-reachable from *p* in terms of *Eps* and *MinPts* in order to discover a cluster. This method produces a cluster in

terms of *Eps* and *MinPts* if *p* is a core point. The next point in the database is visited by DBSCAN if *p* is a border point and no points are density-reachable from *p*. Here is the DBSCAN algorithm:

**Algorithm: DBSCAN**

**Input:**

- D: a data set containing n objects
- ϵ : the radius parameter, and
- MinPts : the neighborhood density threshold

**Output:** A set of density-based clusters.

**Method:**

1. mark all objects as unvisited;
2. do
3. randomly select an unvisited object *p*;
4. mark *p* as visited;
5. if the ϵ-neighborhood of *p* as at least *MinPts* objects
6. create a new cluster *C*, and add *p* to *C*;
7. Let *N* be the set of objects in the ϵ-neighborhood of *p*;
8. for each point *p′* in N
9. if *p′* is unvisited
10. mark *p′* as visited;
11. if the ϵ-neighborhood of *p'* has at least *MinPts* points, add those points to *N*;
12. if *p′* is not yet a member of any cluster, add *p′* to *C*;
13. end for
14. output *C*;
15. else mark *p* as noise;
16. until no object is unvisited;

If two clusters with differing densities are "near" to one another, DBSCAN may combine them into a single cluster. The distance between two sets of points $S_1$ and $S_2$ be characterized as $dist(S_1, S_2) = min\{ dist(p,q) \mid p \in S_1, q \in S_2 \}$. Then, two groups of points with roughly the same density as the thinnest cluster will only be separated from one another if the separation is more than *Eps*. Therefore, for the distinct clusters with

a larger value for *MinPts*, a recursive call to DBSCAN may be crucial. However, DBSCAN's recursive application results in a comprehensive and remarkably potent core method [12].

### 3.2.2 Determining Parameters *Eps* and *MinPts*

This section promotes a simple but effective heuristic to choose the parameters *Eps* and *MinPts*. The following observation provides the foundation for this heuristic. If *d* is an object's separation from its $k^{th}$ nearest neighbor, then practically every object *b*'s d-neighborhood contains exactly *k+1* items. Only if many objects have exactly the same *d* distance from *b*, which is highly improbable, do they all belong to the d-neighborhood of *b*, which contains more than *k+1* items. Furthermore, there aren't significant changes in *d* when chaining *k* for an item in a cluster. This can only occur if an object in a cluster's $k^{th}$ nearest neighbors, where *k* is one, two, three, or more, are roughly positioned on a straight line. Each object is mapped to the distance from its $k^{th}$ nearest neighbor for a specified *k* using the k-distance function, which converts database values to real numbers. The graph of this function provides some information about the density distribution of database when the database's objects are sorted according to their k-distance values. This graph refers to the k-distance graph. All objects with a k-distance value of equal or less will be core objects if an arbitrary object *b* chooses, the parameters *Eps* and *MinPts* are set to (*b*) and (*k*), respectively.

A sorted 4-distance graph is displayed in Figure 3.3. The first item in the first "valley" of the sorted k-distance graph is the threshold point. All objects (left of the threshold) with a larger k-distance value are regarded as noise, while all other points (right of the threshold) are allocated to a cluster. In general, it is highly challenging to mechanically identify the first "valley," yet a user can easily identify this valley in a pictorial representation. Eps and *MinPts* are two required parameters for DBSCAN. The k-distance graph for $k > 4$ does not deviate much from the 4-distance graph, but they also need a lot more processing [12]. Therefore, by setting the parameter *MinPts* to 4 for all databases, the parameter is deleted (for 2-dimensional data).

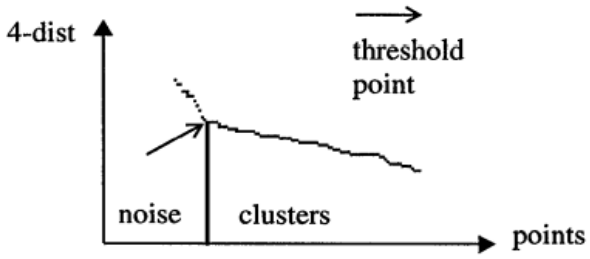


**Figure 3.3 Sorted 4-dist Graph**

### 3.2.3 Density-Based Concepts

The essential **thoughts** of density-based clustering include various definitions, for example, directly accessible, reachable, and connected to densities.

- **Directly density-reachable:** A point *p* is straightforwardly density-reachable from a point *q*. *Eps*, *MinPts* if $p \in N_{Eps}(q)$ and $|N_{Eps}(q)| \geq MinPts$ (core point condition). For sets of core points, it is evident that straightforwardly density-reachable is symmetric. However, it is not often symmetric if both a core point and a border point are present [12]. The directly density-reachable is illustrated in Figure 3.4.

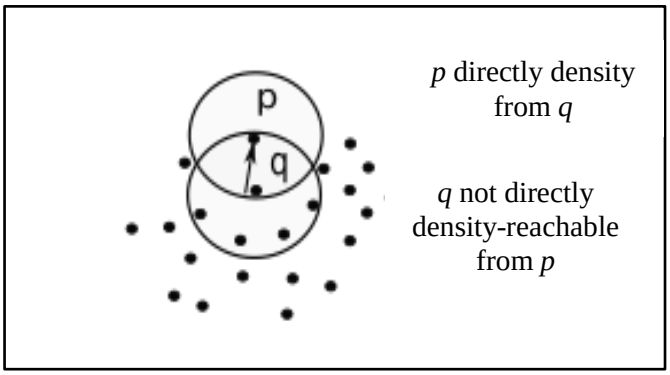


**Figure 3.4 Directly Density-reachable**

- **Density-reachable:** A point *p* is density-reachable from a point *q*. *Eps* and *MinPts* if there is a chain of points $p_1, ..., p_n$, $p_1 = q$, $p_n = p$ such that $p_{i+1}$ is directly density-reachable from $p_i$. The term "density-reachability" is a common expansion of "direct density-reachability." Despite not being symmetric, this relationship is transitive. Density-reachability is undoubtedly symmetric for core points even though it is not as a rule. Because the core point criterion might not apply for both of them, two border points of the same cluster *C* may not be densely reachable from one another. Both of the border points of

*C* must be density-reachable from one of the core points of *C* [12]. Figure 3.5 illustrates density-reachable.

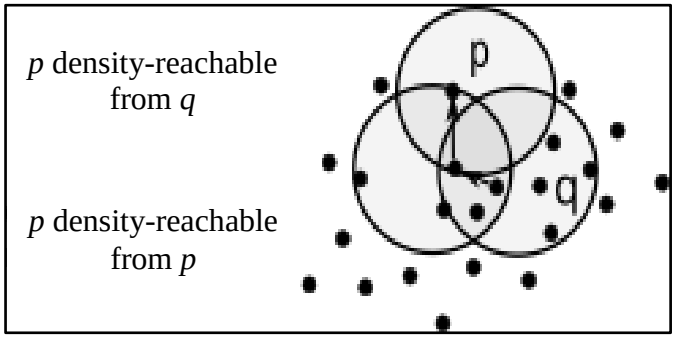


**Figure 3.5 Point *p* and *q* are Density-reachable**

- **Density-connected:** Density links *p* and *q* are present. *p* and *q* are both density-reachable from a point *o* if and only if *Eps* and *MinPts* exist. As a symmetric link, density-connectivity exists. The relationship between density and connectedness is also reflexive for places with a high density of reach. A cluster is automatically thought of as a maximally dense collection of points that are related to one another. In comparison to a specific collection of clusters, noise will be described. A group of points in *D* that do not belong to any of its clusters is called noise [12]. Point *p* and *q* are density-connected to point *o* as shown in Figure 3.6.

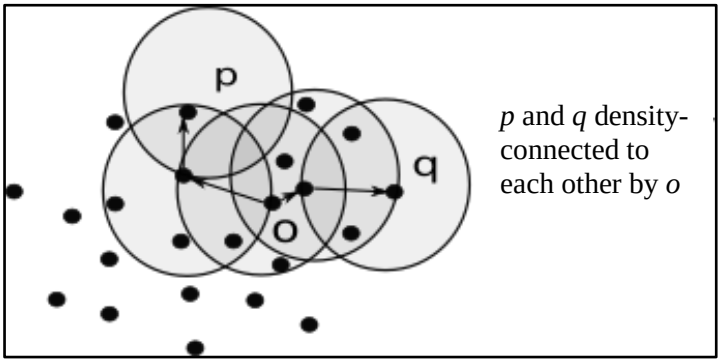


**Figure 3.6 Sample of Density Connected**

### 3.2.4 Explanation of DBSCAN Processing Steps

The following is a description of the DBSCAN processing steps:

- Epsilon (Eps) and minimum points (MinPts) are the two parameters that DBSCAN needs. It starts at a randomly chosen, unexplored starting point.

The next step is to locate every neighboring point that is located within *Eps* of the starting position.

- The formation of a cluster depends on whether the number of neighbors is greater than or equal to *MinPts*. The initial stage and its surrounding stages are added to this cluster, and the initial point is designated as visited. The evaluation procedure is then iterated over for each neighbor in the algorithm.
- The point is designated as noise if the number of neighbors is fewer than *MinPts*.
- The technique then iterates through the remaining unvisited points in dataset when a cluster has been fully enlarged (all points within reach have been visited) [13].

### 3.2.5 Advantages of DBSCAN

Advantages of DBSCAN are as follows:

- Unlike k-means, DBSCAN does not require to know in advance how many clusters there are in the data.
- DBSCAN has the ability to locate clusters of any shape. In fact, it might discover clusters that are entirely encircled by (but unconnected to) another cluster. The so-called single-link effect, in which many clusters are connected by a narrow line of points, is diminished as a result of the *MinPts* parameter.
- Noise exists according to DBSCAN.
- DBSCAN just needs two parameters and is largely unaffected by how the points are arranged in the dataset [13].

### 3.2.6 Disadvantages of DBSCAN

Disadvantages of DBSCAN are as follows:

- When there are significant density discrepancies between the datasets, DBSCAN struggles to properly identify the MinPts and Eps combination for each cluster.
- Choosing a substantial *Eps* figure can be challenging if the data is not properly understood.
- Sensitive to clustering parameters *MinPts.*

- There is some determinism in DBSCAN. This is as a result of the algorithm's random starting point. As a result, border sites that may be reached from different clusters can belong to either cluster [13].

### 3.2.7 K-Distance Graph

The drawbacks of DBSCAN include, if the data is not well understood, picking an eps figure that means something can be challenging. As a result, user select the *Eps* by using the K-distance graph.

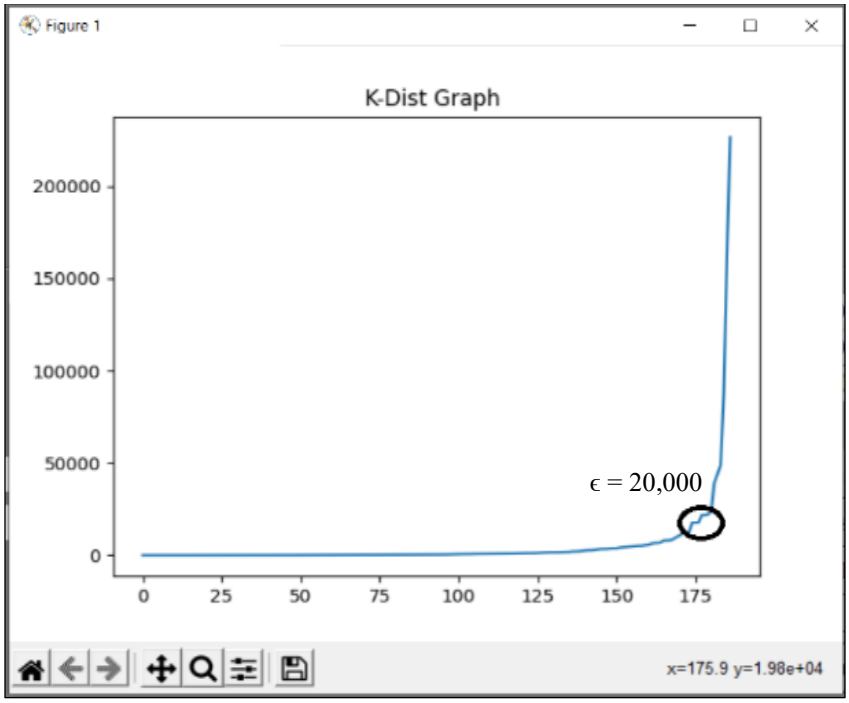


Figure 3.7 The Optimal ϵ Value from K-distance graph

## 3.3 Types of Data in Cluster Analysis

The types of data that cluster analysis typically encounters and how to prepare them for a cluster analysis. Assume that a dataset that needs to be clustered contains *n* items, which could be people, homes, papers, nations, etc. The two data structures shown below are usually used by main memory-based clustering methods.

Data matrix (or object-by-variable structure): This addresses n objects, like people, with p variables (also called measurements or attributes), such as age, height, weight, gender, race, and so on. The structure is in the form of a relational table, or n-by-p matrix (n objects*p variables):

$$\begin{bmatrix} x_{l1} & \cdots\cdots & x_{lf} & \cdots\cdots & x_{lp} \\ \vdots & & & & \vdots \\ x_{i1} & \cdots\cdots & x_{if} & \cdots\cdots & x_{ip} \\ \vdots & & \vdots & & \vdots \\ x_{n1} & \cdots\cdots & x_{nf} & \cdots\cdots & x_{np} \end{bmatrix}$$

Dissimilarity matrix (or object-by-object structure): This stores a collection of proximities that are accessible for all pairs of n objects. It is often represented by an n-by-n table.

$$\begin{pmatrix} 0 & & \\ d\,(2,1) & 0 & \\ d\,(3,1) & d\,(3,2) & 0 \\ . & . & . \\ . & . & . \\ d\,(n,1) & d\,(n,2)\ldots\ldots\ldots\ldots\ldots & 0 \end{pmatrix}$$

Where *d(i, j)* is tithe measured difference or dissimilar between objects *i* and *j*. In general, *d(i, j)* is a nonnegative number that is close to 0 when objects *i* and *j* are highly similar or "near" each other, and becomes larger the more they differ. Since *d(i, j) = d (j, i) = 0*.

Since the rows and columns of the data matrix represent multiple entities while those of the dissimilarity matrix represent the same item, the data matrix is frequently referred to as a two-mode matrix. A dissimilarity matrix is the foundation of many clustering methods. Before using such clustering techniques, the data that are supplied as a data matrix might be converted into a dissimilarity matrix.

### 3.3.1 Interval-Scaled Variables

Continuous measurements of a usually direct scaled are interval-scaled variables. Examples include height and weight, coordinates for latitude and longitude, and temperature. The clustering analysis can be affected by the estimation unit used. In particular applications, the distance between each pair of objects is often used to calculate the dissimilarity (or similarity) between the items represented by interval-scaled variables. In general, smaller unit expressions of variables result in bigger ranges for those variables and, consequently, larger effects on the resulting clustering structure. The data have to be normalized in order to lessen reliance on the measuring units chosen. Attempts to equalize the weight of all variables are made while standardizing

measurements. When there is no prior understanding of the data, this is particularly useful. Users might purposefully need to give a particular set of variables in some apps more weight than they do in others. Regular processing is done based on the distance between each pair of items to determine how distinct or similar the objects described by interval-scaled variables are [6].

### 3.4 Similarity or Distance Measurements

Similarity or distance functions are essential to all clustering techniques. Establishing the similarity or distance matrix is the first stage in cluster analysis. This matrix is a table where the rows and columns serve as the analytical units and the cell entries serve as a gauge of how similar or dissimilar any two situations are. The distance functions for numerical qualities are Manhattan (City Block) distance and Euclidean distance. The Minkowski distance is a more general distance function that includes both distance measurements as special cases.

Usually, the distance between each pair of objects is used to calculate how unlike or similar two objects are. Three popular distance calculation methods are described below: Manhattan distance, which is described as

$$d(a,b) = |x_{a1} - y_{b1}| + |x_{a2} - y_{b2}| + \cdots + |x_{an} - y_{bn}| \quad (3.1)$$

Euclidean distance measure is defined as

$$\mathrm{d(a,b)} = \sqrt{(\mathrm{x_{a1}} - \mathrm{x_{b1}})^2 + (\mathrm{x_{a2}} - \mathrm{x_{b2}})^2 + .. + (\mathrm{x_{an}} - \mathrm{x_{bn}})^2} \quad (3.2)$$

Minkowski distance method is defined as

$$\mathrm{d(a,b)} = \sqrt[h]{|x_{a1} - x_{b1}|^k + |x_{a2} - x_{b2}|^k + \cdots + |x_{an} - x_{an}|^k} \quad (3.3)$$

Where $k = 3$

The formula defines "*a*" and "*b*" as data items with *n* dimensions each. The formula above uses the following notation to indicate the distance between the two data objects, *d(a, b)*. The measurement of item "*a*" in dimension "*n*" is the $x_{an}$. The following mathematical conditions for a distance function are met by distance methods:

- *d(a, b) >= 0*: Distance is a nonnegative number.
- *d(a, b) = 0*: The distance of an object to itself is 0.

- *d(a, b) = d(b, a)*: Distance is a symmetric function.
- *d(a, b) <= d(a, h) + d(h, b)*: Going directly from object *a* to *b* in space is no more than making a detour over any other object *h* (triangular inequality)[6].

### 3.5 Silhouette Coefficient

A metric used to assess the effectiveness or quality of clusters is the silhouette coefficient, which is based on the cohesion and separation of each sample point. The "*a"* is the average intra-cluster distance, which is determined by averaging the distances between each object in the cluster to which an object belongs. The "*b"* is the minimal average distance between all clusters to which "*o"* does not belong. It also refers to the average inter-cluster distance.

*a* = the average distance in cluster *A*

*b* = minimum of average distance all clusters *(B, C)*

$$\textbf{Silhouette Score} = \frac{b-a}{max\{a,b\}} \tag{3.4}$$

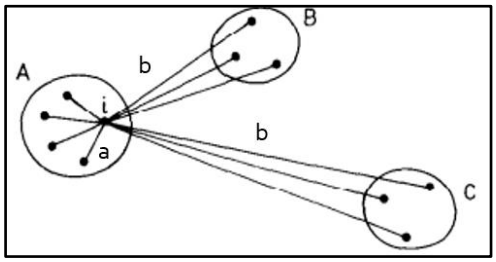


**Figure 3.8: Calculate Silhouette Score on Clusters**

**-1 ≤ Silhouette Score ≤ 1**

The range of the silhouette scores is -1 to 1. When the value is near to 1, the cluster members are clearly distinct from one another and are spaced widely apart. Close to 0 are indifferent or have negligible distances between them. If the value is close to -1, clusters have been incorrectly assigned [1].

# CHAPTER 4

# IMPLEMENTATION OF THE SYSTEM

This chapter describes the proposed system design and system flow diagram about clustering of countries based on the number of COVID-19 cases. Then, the data normalization, attributes and its information about COVID-19 cases are presented. The detail explanation, implementation and experimental results of the system are also described in this chapter.

## 4.1 Proposed System Architecture

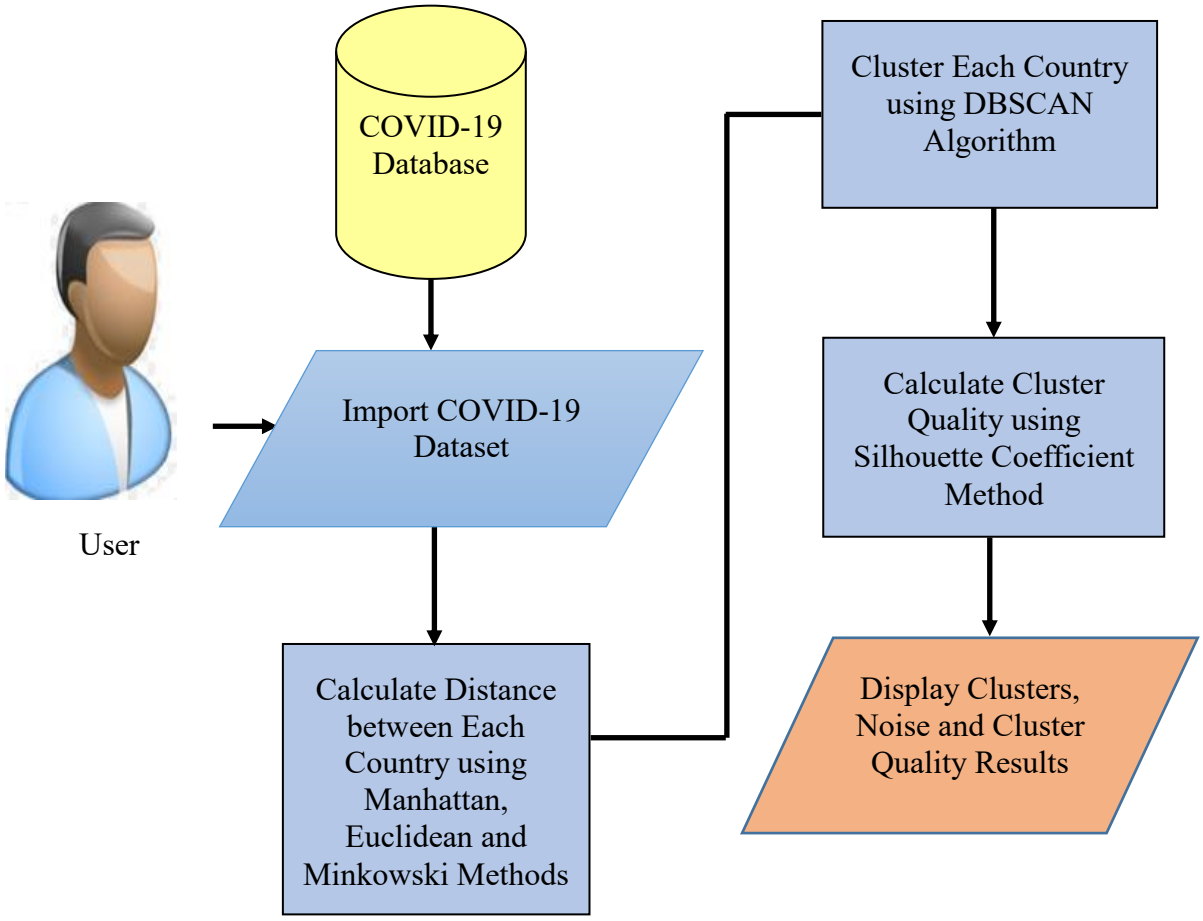


**Figure 4.1 Proposed System Design**

The proposed system design is shown in Figure 4.1. By using DBSCAN algorithm, this system is implemented as the countries clustering system based on the number of COVID-19 cases such as confirmed cases, active cases, death cases, recovered cases, active cases, new cases, new death, and new recovered. Firstly, the

system imports statistical COVID-19 dataset and normalizes it by the population of 10 million people in each country due to the different population size. Then, the system calculates the similarities or distances between each country by using three distance measuring methods. Based on the distance results, each country is clustered by using DBSCAN algorithm. After clustering process, the Silhouette Coefficient value is calculated to measure the quality of each cluster. Finally, this system displays the cluster results, noise and clustering performance.

## 4.2 Process Flow of the System

In this proposed system, the user firstly imports statistical COVID-19 cases dataset for DBSCAN process. Then, the user inputs the minimum number of points (MinPts) which is the nearest neighbor value. To calculate the distance between each object, user can choose the distance measuring methods, namely,

- Euclidean distance,
- Manhattan distance and
- Minkowski distance.

According to the user selected distance measuring method, system calculates the distance or similarities between each object (each country) based on numbers of confirmed cases, death cases, recovered cases, active cases, new cases, new death cases and new recovered cases. Then, draws the *k* distance graph of selected distance measure. After that, identifies the $\epsilon$ value based on the resultant *k* distance graph. The $\epsilon$ value corresponds to critical change or strong bended point in curves on *k* distance graph. By using the given $\epsilon$ and *MinPts* values, the core and border objects are identified. According to the DBSCAN process, this system clusters each core object until all objects have been processed. If an object is a core object, the DBSCAN determines all points expect this point is density-reachable from it and forms a cluster. After clustering each core object, this system identifies noise object that is not included in some clusters. After finishing the clustering process, the Silhouette score is calculated to measure the resultant clustering quality. The cluster quality results are different based on the selected similarities calculation methods and $\epsilon$ value. This system compares the clustering quality of each country using silhouette scores. Finally, this system displays the groups of countries (clusters), noises and cluster quality results. The system flow diagram is illustrated in Figure 4.2

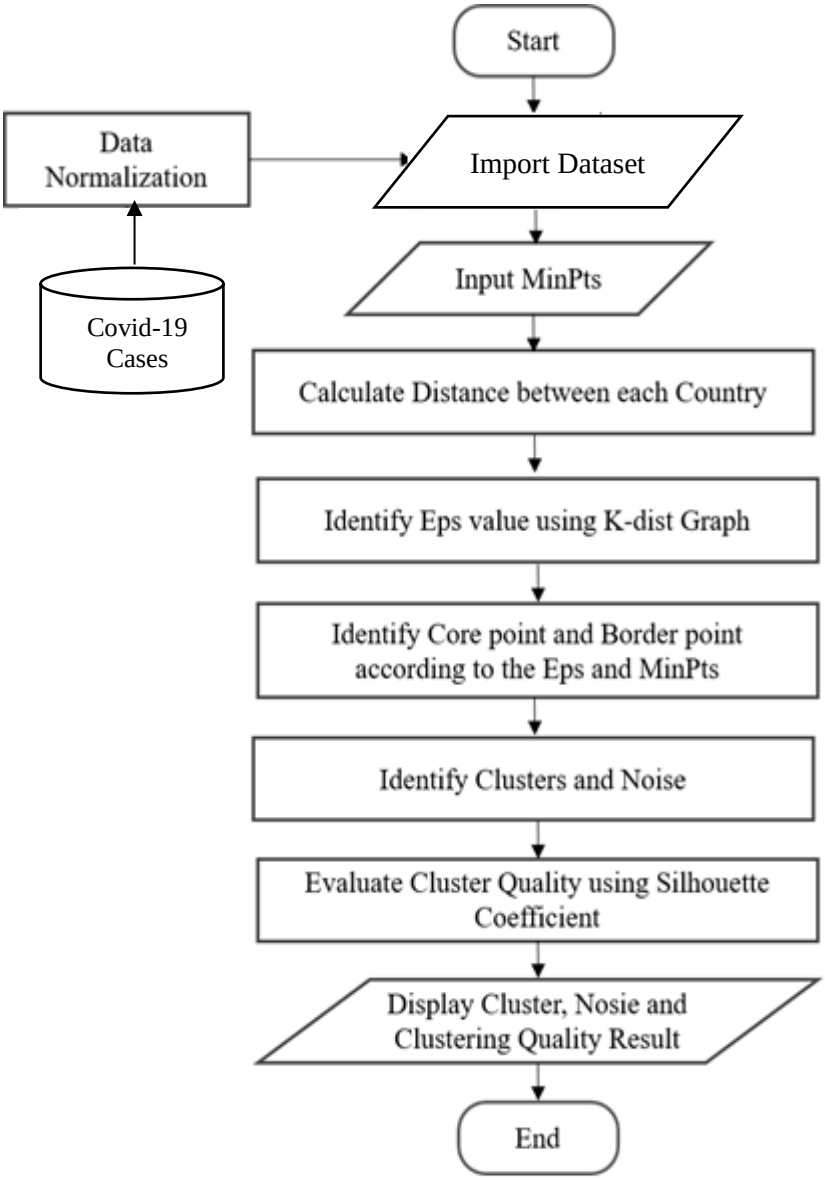


**Figure 4.2 Process Flow Diagram of the System**

## 4.3 Database and its Attribute Information of the System

The statistical coronavirus COVID-19 dataset is extracted from the data source www.kaggle.com. This dataset includes the “35,157” records that are collected from the date of “22/1/2020” to “27/7/2020”. These statistics data are based on “187” countries under the WHO region. Database and its attribute information are shown in Table 4.1.

**Table 4.1 Database and its Attribute Information**

| ID | Attribute Name | Attribute Information |
|---|---|---|
| 1 | Country | Country name under WHO (world health organization) region. |
| 2 | Confirmed | Number of COVID-19 confirmed cases that suffered in each country. |
| 3 | Deaths | Number of deaths cases that faced in a country. |
| 4 | Recovered | Number of recovered cases from the COVID-19 confirmed cases that caused in a country. |
| 5 | Active | Number of active COVID-19 cases within one day that causes in a country. |
| 6 | New cases | Number of new COVID-19 cases that caused in each country. |
| 7 | New deaths | Number of new deaths cases from the COVID-19 active cases that caused in each country. |
| 8 | New recovered | Number of new recovered cases from the COVID-19 active cases that caused in each country. |

## 4.4 Explanation of the System

The statistical COVID-19 cases dataset that are extracted from data source represents the absolute number of confirmed cases, deaths cases, recovered cases, active cases, new cases, new deaths and new recovered cases. Since there is a huge difference in population size and also in number of tested cases among countries, this system is not considered on the original absolute numbers of COVID-19 cases. For that reason, data are considered on the number of reported COVID-19 cases per 10 million people for each nation. **Rate of Cases** from each country are calculated based on 10 million by using the following equations.

$$\text{Rate of Cases (country)} = \frac{cases}{papulation} * 10million$$

Last but not least, the dataset were used in this study, which includes the cumulative number taken for each country and every day. As a sample, this system is evaluated on the date 1/5/2020. The first 10 statistic records are shown in Table 4.2.

**Table 4.2 COVID-19 Statistics Data**

| Country | Confirmed Cases | Deaths Cases | Recovered Cases | Active Cases | New Cases | New Deaths | New Recovered |
|---|---|---|---|---|---|---|---|
| Afghanistan | 602.6 | 17.5 | 80.0 | 505.1 | 42.3 | 1.0 | 12.9 |
| Albania | 2716.7 | 107.7 | 1695.3 | 913.7 | 31.3 | 0 | 62.5 |
| Algeria | 950.8 | 103.7 | 416.8 | 430.4 | 33.9 | 0.7 | 9.6 |
| Andorra | 96452.6 | 5567.0 | 60590.4 | 30295.2 | 0 | 129.5 | 0 |
| Angola | 9.2 | 0.61 | 3.37 | 5.2 | 0.92 | 0 | 1.2 |
| Antigua | 2557.2 | 306.9 | 1534.3 | 716.0 | 102.2 | 0 | 409.1 |
| Argentina | 1004.6 | 49.9 | 286.4 | 668.4 | 23.1 | 1.6 | 7.9 |
| Armenia | 7251.5 | 111.4 | 3298.3 | 3841.8 | 276.8 | 3.4 | 162.0 |
| Australia | 2664.4 | 36.6 | 2270.1 | 357.7 | 4.7 | 0 | 12.9 |
| Afghanistan | 602.7 | 17.6 | 80.01 | 505.1 | 42.3 | 1.0 | 12.9 |
| … | … | … | .. | … | … | .. | … |

After normalizing the statistical COVID-19 cases dataset based on countries population, the user inputs the *MinPts* value. In this example, *MinPts = 3* is inputted from the user. Then, the user can choose either the Euclidean or Manhattan or Minkowski to calculate the distances between points. After the user has chosen, this system calculates the distance between each data record. Then, find the *k* distance metric between each record. One record represents one WHO country. Table 4.3 shows the distance matrix using Manhattan method. Table 4.4 and 4.5 show the distance matrix of Euclidean and Minkowski methods.

**Table 4.3 Distance Matrix using Manhattan Method**

| | Afghanistan | Albania | --- | Yemen | Zambia | Zimbabwe |
|---|---|---|---|---|---|---|
| Afghanistan | 0 | 4289.877 | --- | 1256.598 | 1130.311 | 1207.664 |
| Albania | 4289.877 | 0 | --- | 5522.285 | 5395.998 | 5473.351 |
| Algeria | 858.03 | 3587.985 | --- | 1940.896 | 1816.177 | 1891.962 |
| Andorra | 191883.519 | 187695 | --- | 193030.3 | 192927.4 | 192980.7 |
| Angola | 1241.126 | 5506.813 | --- | 15.594 | 110.815 | 37.75 |

| | Afghanistan | Albania | --- | Yemen | Zambia | Zimbabwe |
|---|---|---|---|---|---|---|
| --- | --- | --- | --- | --- | --- | --- |
| West Bank and Gaza | 244.436 | 4125.767 | --- | 1396.518 | 1291.027 | 1347.584 |
| Western Sahara | 1067.963 | 5325.39 | --- | 198.917 | 99.338 | 161.541 |
| Yemen | 1256.598 | 5522.285 | --- | 0 | 126.287 | 49.608 |
| Zambia | 1130.311 | 5395.998 | --- | 126.287 | 0 | 86.283 |
| Zimbabwe | 1207.664 | 5473.351 | --- | 49.608 | 86.283 | 0 |

**Table 4.4 Distance Matrix using Euclidean Method**

| | Afghanistan | Albania | --- | Yemen | Zambia | Zimbabwe |
|---|---|---|---|---|---|---|
| Afghanistan | 0 | 2693.8 | --- | 789.15 | 732.21 | 757.5852632 |
| Albania | 2693.762 | 0 | --- | 3330.1 | 3258.4 | 3303.136912 |
| Algeria | 497.7812 | 2233.7 | --- | 1126.5 | 1057.3 | 1097.088176 |
| Andorra | 117332.7 | 114665 | --- | 117994 | 117922 | 117966.92 |
| Angola | 781.1365 | 3322 | --- | 8.5221 | 64.511 | 23.88610387 |
| --- | --- | --- | --- | --- | --- | --- |
| West Bank and Gaza | 121.8224 | 2578.1 | --- | 887.19 | 827.03 | 855.6722293 |
| Western Sahara | 701.7389 | 3203 | --- | 130.34 | 61.048 | 109.6403928 |
| Yemen | 789.1492 | 3330.1 | --- | 0 | 72.575 | 31.6776323 |
| Zambia | 732.2143 | 3258.4 | --- | 72.575 | 0 | 50.67547238 |
| Zimbabwe | 757.5853 | 3303.1 | --- | 31.678 | 50.675 | 0 |

**Table 4.5 Distance Matrix using Minkowski Method**

| | Afghanistan | Albania | --- | Yemen | Zambia | Zimbabwe |
|---|---|---|---|---|---|---|
| Afghanistan | 0 | 2394.7 | --- | 701.22 | 651.25 | 673.026 |
| Albania | 2394.702 | 0 | --- | 2948.4 | 2885 | 2924.688 |
| Algeria | 433.539 | 1975.5 | --- | 1001.9 | 940.79 | 975.749 |
| Andorra | 104119.9 | 101750 | --- | 104698 | 104634 | 104674.036 |
| Angola | 694.185 | 2941.2 | --- | 7.407 | 56.701 | 21.204 |
| --- | --- | --- | --- | --- | --- | --- |
| West Bank and Gaza | 103.175 | 2292.8 | --- | 786.61 | 734.32 | 758.419 |
| Western Sahara | 623.855 | 2835.8 | --- | 115.77 | 53.701 | 97.677 |
| Yemen | 701.223 | 2948.4 | --- | 0 | 63.675 | 28.226 |
| Zambia | 651.248 | 2885 | --- | 63.675 | 0 | 44.336 |
| Zimbabwe | 673.026 | 2924.7 | --- | 28.226 | 44.336 | 0 |

After calculating distance between each object, the user identifies the $\epsilon$ value through the K-distance graph from three distance methods. The optimal Epsilon value ($\epsilon$) is corresponding to the critical change (strong bend) in curves on K-distance graph. Therefore, user choose Epsilon value is 11,000 for Euclidean distance. Figure 4.3 shows the K-distance graph using Euclidean distance measure.

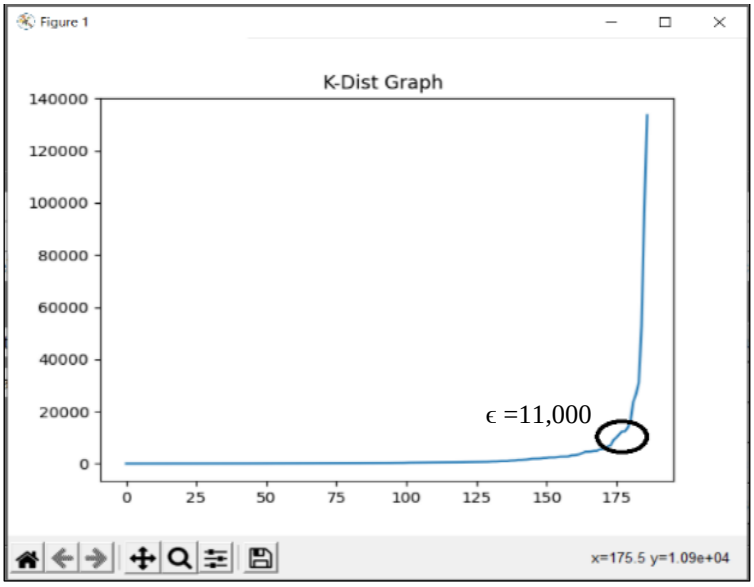


**Figure 4.3 K-Distance Graph for Euclidean Distance Method**

As the same way, this system finds the K-distance graph and choose the ϵ value for other distance measures: Manhattan and Minkowski as shown in Figure 4.4 and 4.5. The value 20,000 is selected from Manhattan distance graph and 9,000 is used for Minkowski distance as shown in Table 4.6.

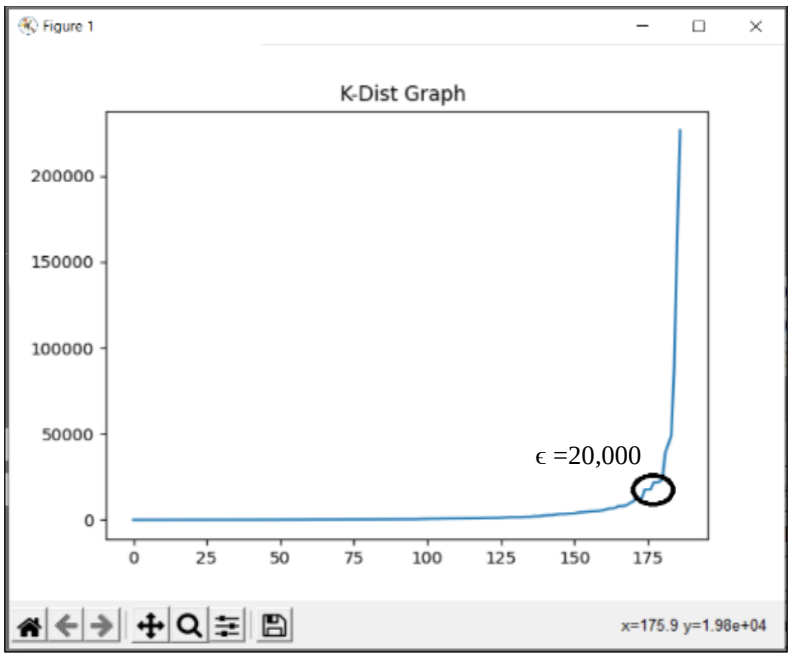


**Figure 4.4 K-Distance Graph for Manhattan Distance Method**

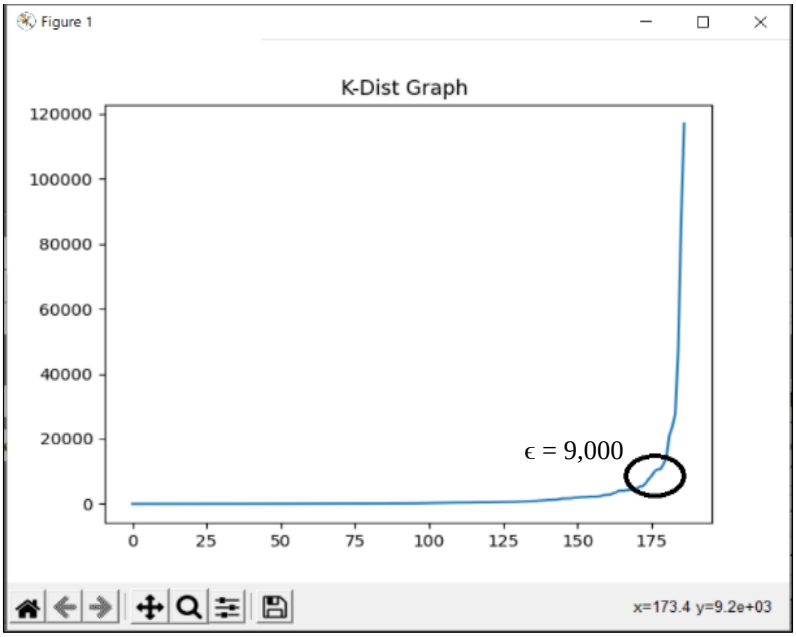


**Figure 4.5 K-Distance Graph for Minkowski Distance Method**

**Table 4.6 Epslion (ϵ) Values and MinPts**

| Distance Measure | MinPts | Epslion (ϵ) |
|---|---|---|
| Euclidean Distance | 3 | 11000 |
| Manhattan Distance | 3 | 20000 |
| Minkowski Distance | 3 | 9000 |

After identified two important parameters that are required for DBSCAN, the system identifies the core objects from all objects. Core object is satisfied with the user defined MinPts and Epslion (ϵ) values. Table 4.7, 4.8 and 4.9 shows the core objects based on Manhattan, Euclidean and Minkowski distance methods.

**Table 4.7 Core Points Result based on Manhattan Distance**

| Index | Core object | Country |
|---|---|---|
| 0 | Core | Afghanistan |
| 1 | Noise | Albania |
| 2 | Noise | Algeria |
| 3 | Noise | Andorra |
| 4 | Core | Angola |
| --- | --- | --- |
| 182 | Core | West Bank and Gaza |
| 183 | Core | Western Sahara |

| Index | Core object | Country |
|---|---|---|
| 184 | Core | Yemen |
| 185 | Core | Zambia |
| 186 | Core | Zimbabwe |

**Table 4.8 Core Points Result based on Euclidean Distance**

| Index | Core object | Country |
|---|---|---|
| 0 | Core | Afghanistan |
| 1 | Noise | Albania |
| 2 | Noise | Algeria |
| 3 | Noise | Andorra |
| 4 | Core | Angola |
| --- | --- | --- |
| 182 | Core | West Bank and Gaza |
| 183 | Core | Western Sahara |
| 184 | Core | Yemen |
| 185 | Core | Zambia |
| 186 | Core | Zimbabwe |

**Table 4.9 Core Points Result based on Minkowski Distance**

| Number | Core object | Country |
|---|---|---|
| 0 | Core | Afghanistan |
| 1 | Noise | Albania |
| 2 | Noise | Algeria |
| 3 | Noise | Andorra |
| 4 | Core | Angola |
| --- | --- | --- |
| 182 | Core | West Bank and Gaza |
| 183 | Core | Western Sahara |
| 184 | Core | Yemen |
| 185 | Core | Zambia |
| 186 | Core | Zimbabwe |

This system forms clusters with core objects using DBSCAN algorithm. Also, this system defines the noise object that does not include in any cluster. With the use of Euclidean distance, the system produced four clusters. In Cluster 1 contains 135

countries, Cluster 2 contains 7 countries, Cluster 3 contains 5 countries, Cluster 4 have 3 countries and 37 countries are noises. Table 4.10 shows the cluster results based on Euclidean distance method.

**Table 4.10 Cluster Results based on Euclidean Distance Method**

| **Noise** | **Cluster 1** | **Cluster 2** | **Cluster 3** | **Cluster 4** |
|---|---|---|---|---|
| Andorra | Afghanistan | Armenia | Belarus | Chile |
| Austria | Albania | Cyprus | Canada | Kuwait |
| Bahrain | Algeria | Czechia | Ecuador | Moldova |
| Belgium | Angola | Dominican Republic | Norway | |
| Denmark | Antigua and Barbuda | North Macedonia | Panama | |
| Djibouti | Argentina | Saudi Arabia | | |
| Estonia | Australia | Slovenia | | |
| Finland | Azerbaijan | | | |
| France | Bahamas | | | |
| Germany | Bangladesh | | | |
| --- | ---- | | | |

While using Manhattan distance, it made four clusters. There are 127 countries in Cluster 1, 8 countries in Cluster 2, 3 countries in Cluster 3, and another 3 countries contains in Cluster 4. This turn, the system gave 36 countries as noises. Table 4.11 shows the cluster results based on Manhattan distance method.

**Table 4.11 Cluster Results based on Manhattan Distance Method**

| **Noise** | **Cluster 1** | **Cluster 2** | **Cluster 3** | **Cluster 4** |
|---|---|---|---|---|
| Andorra | Afghanistan | Armenia | Canada | Chile |
| Austria | Albania | Cyprus | Ecuador | Kuwait |
| Bahrain | Algeria | Czechia | Panama | Moldova |
| Belarus | Angola | Dominican Republic | | |
| Belgium | Antigua and Barbuda | North Macedonia | | |

| | | | | |
|---|---|---|---|---|
| Denmark | Argentina | Romania | | |
| Djibouti | Australia | Saudi Arabia | | |
| Estonia | Azerbaijan | Slovenia | | |
| Finland | Bahamas | | | |
| France | Bangladesh | | | |
| --- | ---- | | | |

Using Minkowski distance method, the system also generated four clusters. Cluster 1 have 127 countries, Cluster 2 have 7 countries, Cluster 3 have 8 countries and Cluster 4 have 4 countries. The rest 41 countries are denoted as noises. Table 4.12 shows the cluster results based on Minkowski distance method.

**Table 4.12 Cluster Results based on Minkowski Distance Method**

| **Noise** | **Cluster 1** | **Cluster 2** | **Cluster 3** | **Cluster 4** |
|---|---|---|---|---|
| Andorra | Afghanistan | Armenia | Bosnia and Herzegovina | Belarus |
| Austria | Albania | Cyprus | Brazil | Canada |
| Bahrain | Algeria | Czechia | Croatia | Ecuador |
| Belgium | Angola | Dominican Republic | Kosovo | Norway |
| Chile | Antigua and Barbuda | North Macedonia | Latvia | Panama |
| Denmark | Argentina | Saudi Arabia | Lithuania | |
| Djibouti | Australia | Slovenia | Montenegro | |
| Estonia | Azerbaijan | | Oman | |
| Finland | Bahamas | | | |
| France | Bangladesh | | | |
| --- | ---- | | | |

## 4.5 Implementation of the System

This system is proposed the clustering of countries based on statistics number of COVID-19 cases by using DBSCAN algorithm. It is implemented by using Python programming language.

### 4.5.1 Welcome Page

First of all, the user must choose the start date and end date to cluster countries. This system allows the user to cluster countries daily basis or monthly or the whole dataset. Then, the user can choose the desired distance calculation method to calculate the distance between each country. These distance measures are Euclidean, Manhattan and Minkowski methods. If the user clicks the "Distance Matrix" button, the system calculates distances between each country and shows the distance results. Welcome page of the system is illustrated in Figure 4.6.

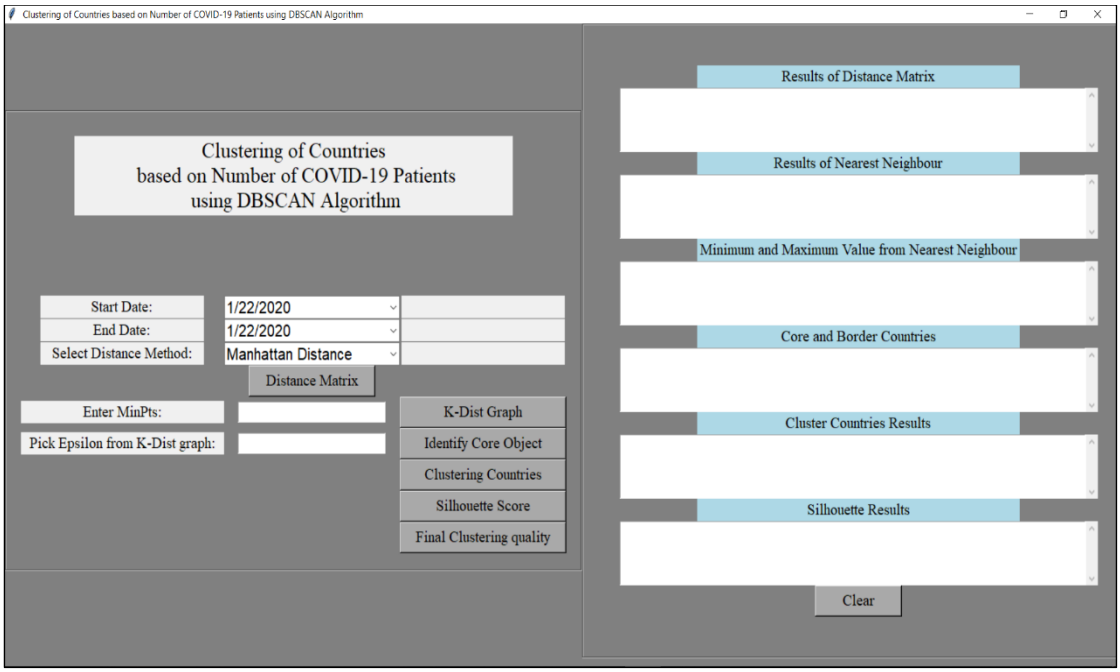


**Figure 4.6 Welcome Page of the System**

Then, the user needs to input the minimum number of points "MinPts" ($2 \leq$ MinPts $\leq 14$) to identify the nearest neighbor countries. As a sample, if the user enters the "MinPts = 3", this system chooses the three nearest neighbor countries from one country. It is needed to pick Epsilon ($\epsilon$) from the K-distance graph to cluster the countries. The user can view the K-distance graph by pressing "K-Dist Graph" button. Based on the *MinPts* and *Epsilon* ($\epsilon$) values, this system identifies core object (core

country) among other countries within WHO region. By using core objects, each country is clustered and also produced the noise object. The user has to click the "Clustering countries" button to view the clustering countries. After clustering process, this system calculates the clustering quality (strength of cohesion and separation) of each country by pressing the "Silhouette Score" button and display silhouette score graph. Finally, the user can view the cluster quality via the "Final Clustering Quality" button.

### 4.5.2 Distance Matrix between Each Country

This system calculates the distance between each country. As a sample, the distance matrix by using Manhattan method is shown in Figure 4.7.

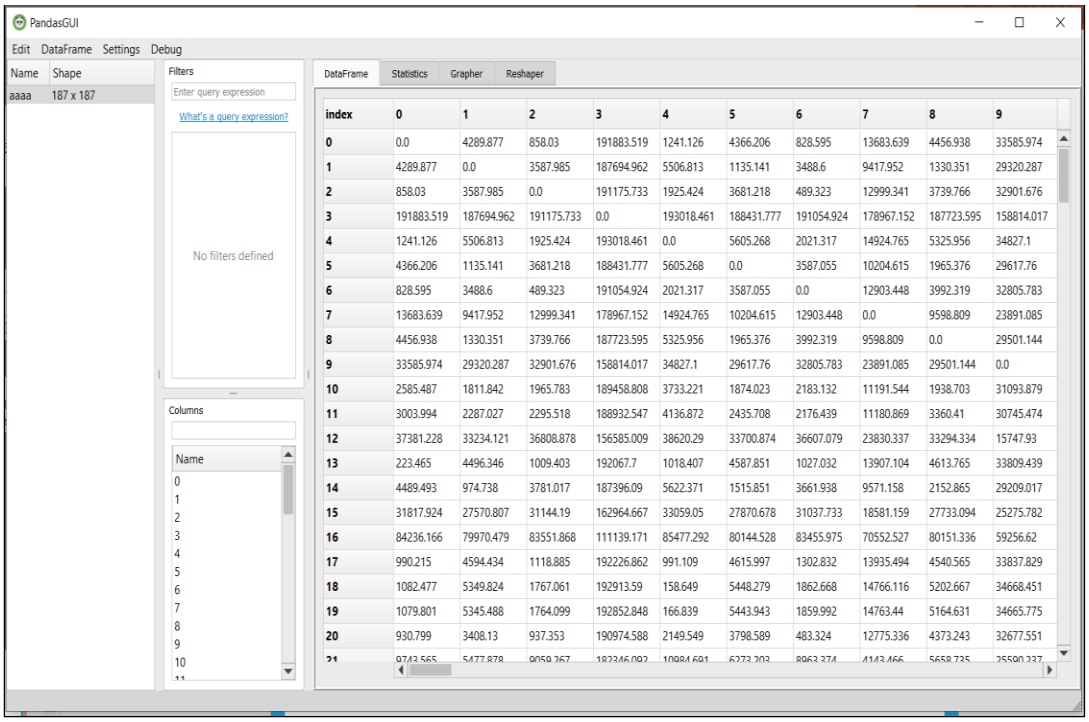


**Figure 4.7 Distance Result using Manhattan Method**

### 4.5.3 Nearest Neighbor Countries

This system searches the nearest neighbor countries from one country by using distance matrix (distance measures table). Nearest neighbors will change according to the user specified "*MinPts*" value. Nearest neighbor countries are shown in Figure 4.8.

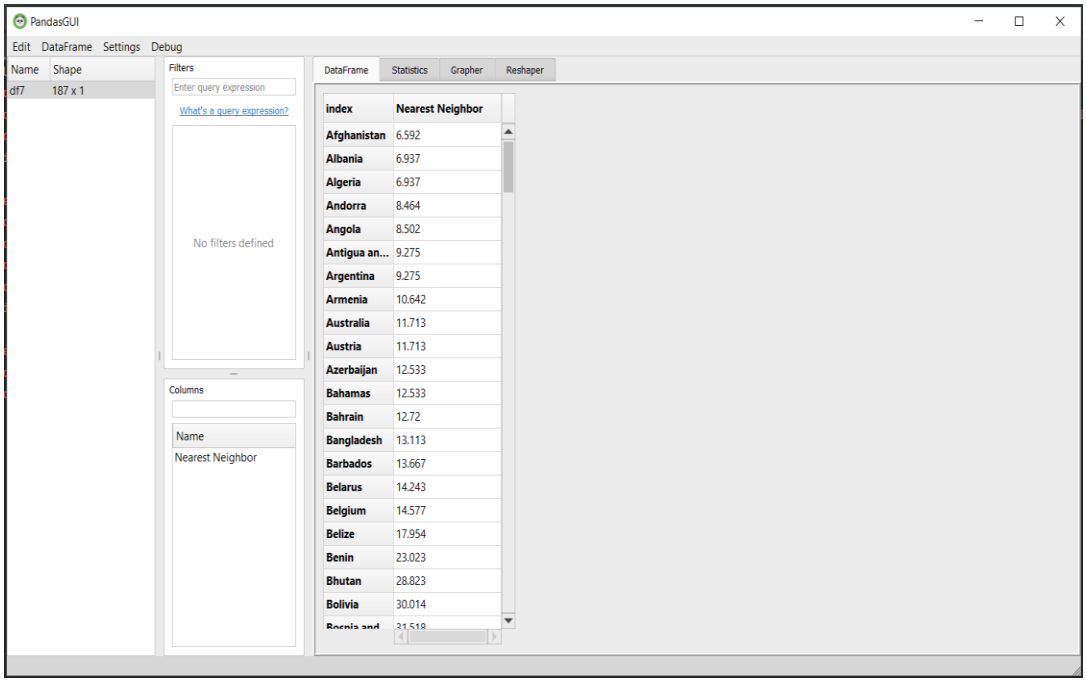


**Figure 4.8 Nearest Neighbor Results**

### 4.5.4 Identifying Core Objects for Clustering

Core objects are essential for DBSCAN clustering algorithm. Core object (country) is the object that satisfies the "*MinPts*" and "ϵ" value. Core object for clustering is depicted in Figure 4.9.

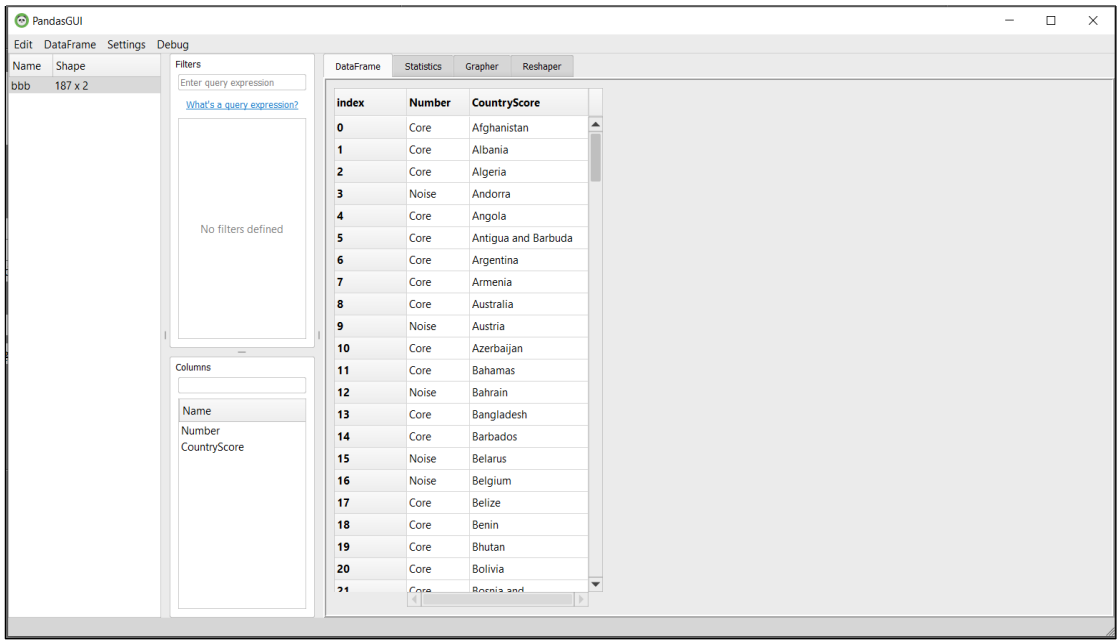


**Figure 4.9 Core Object for Clustering**

### 4.5.5 Countries Clustering Result

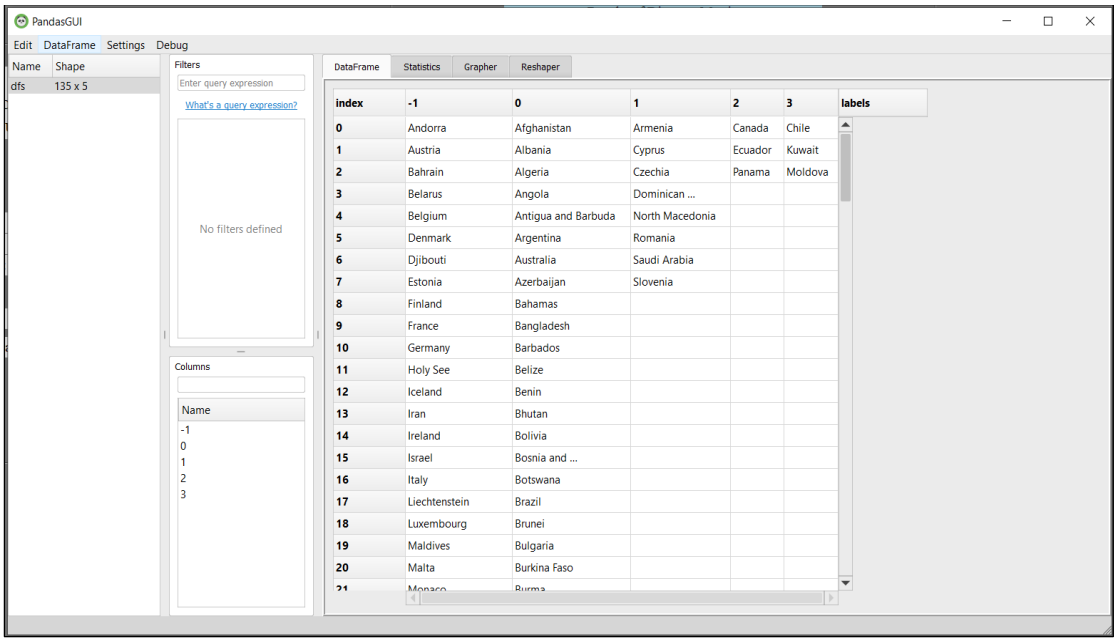


**Figure 4.10 Countries Clustering Result**

This system clusters each core country. If the object (country) that does not contain any cluster, this object is denoted as noise object. In this system, index (-1) represents the "Noise" objects. Index (0) represents cluster 1 and so on. Countries clustering countries result is shown in Figure 4.10.

### 4.5.6 Cluster Qualities using Silhouette Score

After clustering with DBSCAN process, this system calculates the Silhouette score on each cluster. Silhouette scores for each country are illustrated in Figure 4.11 and silhouette score graph is shown in Figure 4.12.

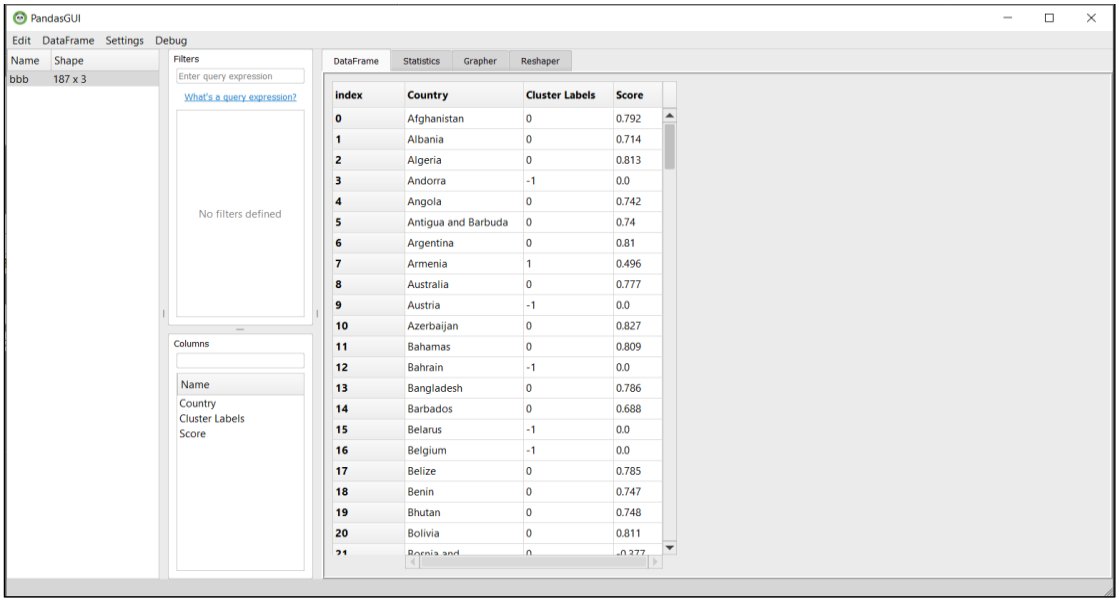


**Figure 4.11 Silhouette Score Countries Result**

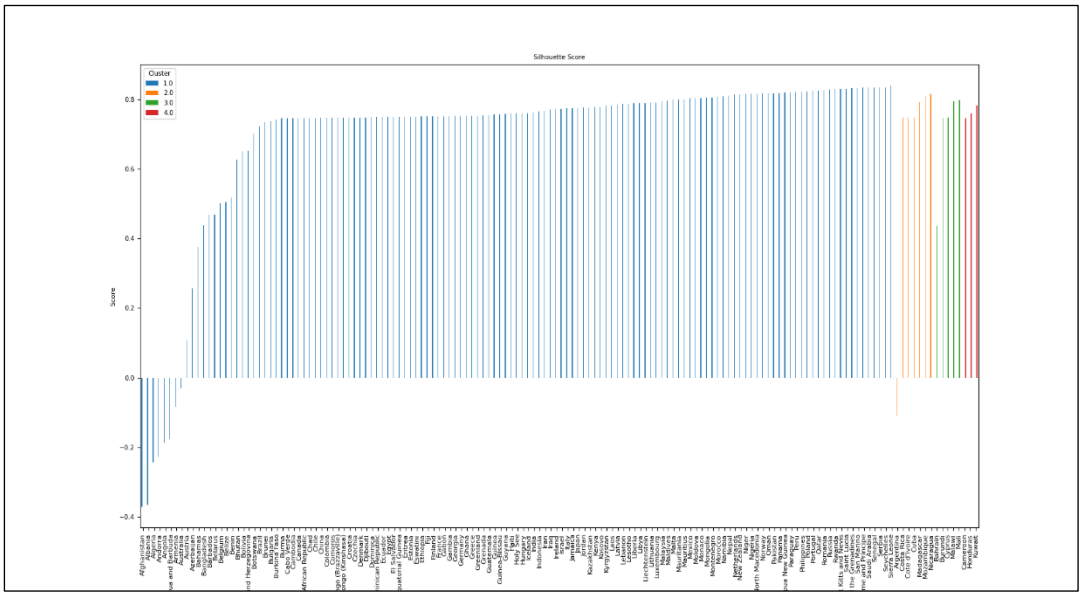


**Figure 4.12 Silhouette Score Graph**

Finally, this system produces average silhouette score and input data from user that is Start Date, End Date, Distance Matrix, MinPts and Epsilon and also produces number of cluster and noise as shown in Figure 4.13.

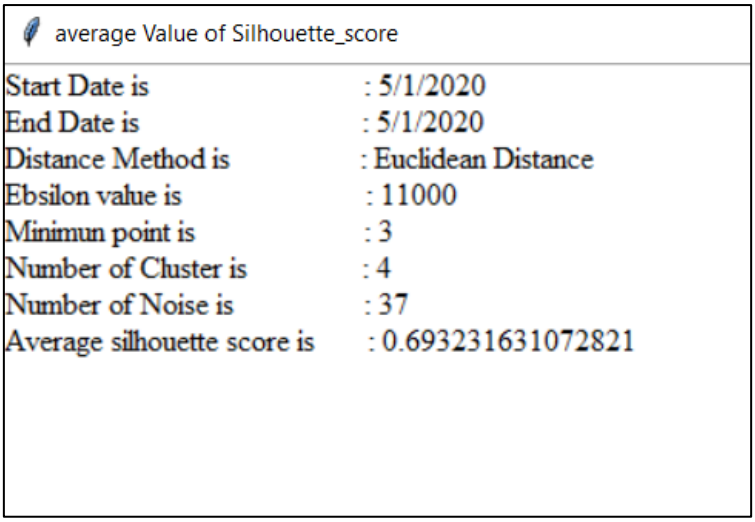


**Figure 4.13: Average Silhouette Score**

## 4.6 Experimental Result of the System

For experimental result of the system, this system calculates the clustering quality by using the Silhouette scores method. Clustering quality can change depending on the user desired distance methods. Table 4.13 shows the silhouette score based on Euclidean method. And, the silhouette score graph using Euclidean is shown in Figure 4.14.

**Table 4.13 Silhouette Score based on Euclidean Method**

| Index | Country | Silhouette Score |
|---|---|---|
| 1 | Afghanistan | 0.79544256 |
| 2 | Albania | 0.7421843 |
| ---- | ---- | ---- |
| 70 | Germany | 0 |
| 71 | Ghana | 0.79989379 |
| 72 | Greece | 0.77571712 |
| 73 | Greenland | 0.83311619 |
| ---- | ---- | ---- |
| 185 | Yemen | 0.834 |
| 186 | Zambia | 0.8340 |
| 187 | Zimbabwe | 0.8398 |

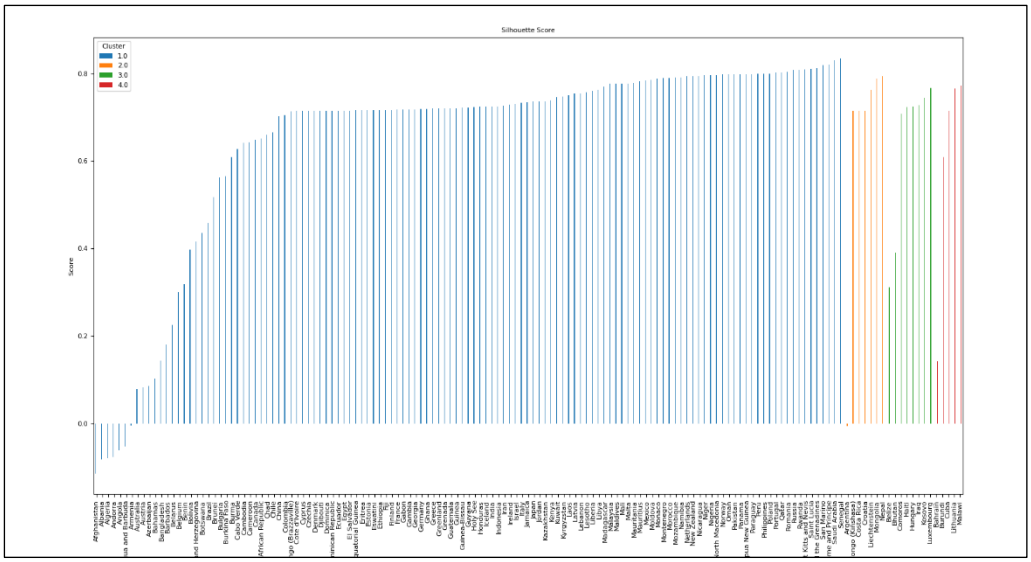


**Figure 4.14 Silhouette Score Graph using Euclidean**

Table 4.14 shows the silhouette score based on Manhattan method. And, the silhouette score graph using Manhattan is illustrated in Figure 4.15.

**Table 4.14 Silhouette Score Result based on Manhattan Method**

| Index | Country | Silhouette Score |
|---|---|---|
| 1 | Afghanistan | 0.79230895 |
| 2 | Albania | 0.71399929 |
| ---- | ---- | ---- |
| 70 | Germany | 0 |
| 71 | Ghana | 0.79701839 |
| 72 | Greece | 0.75489585 |
| 73 | Greenland | 0.84922168 |
| ---- | ---- | ----- |
| 185 | Yemen | 0.8292 |
| 186 | Zambia | 0.8293 |
| 187 | Zimbabwe | 0.8492 |

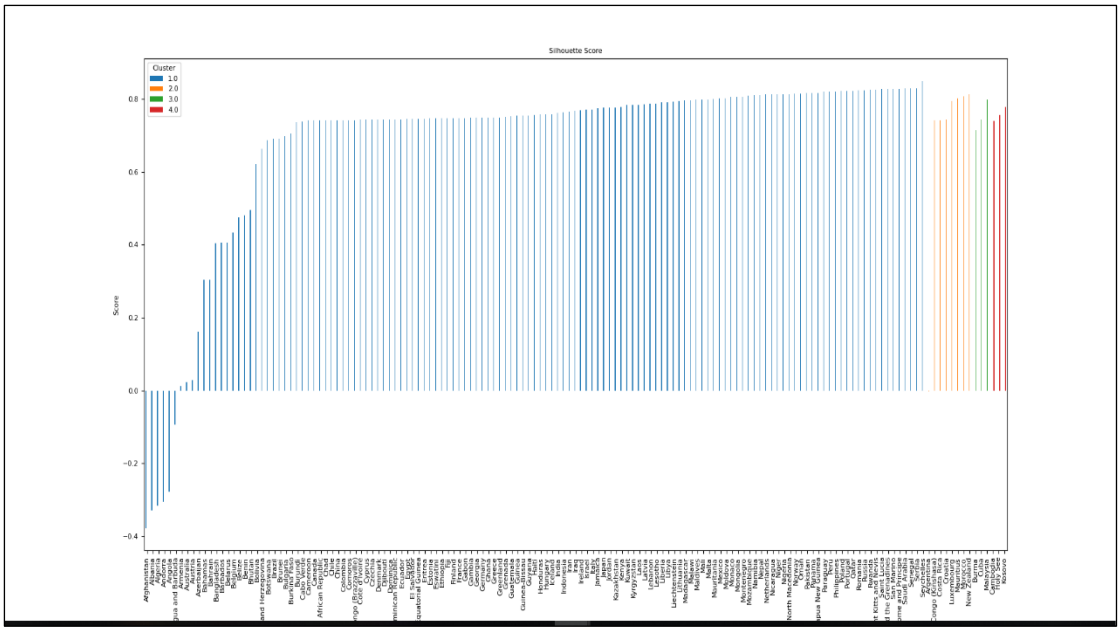


**Figure 4.15 Silhouette Score Graph using Manhattan**

Table 4.15 shows the silhouette score based on Minkowski method. And, the silhouette score graph using Minkowski is depicted in Figure 4.16.

**Table 4.15 Silhouette Score Result based on Minkowski Method**

| ID | Country | Silhouette Score Results |
|---|---|---|
| 1 | Afghanistan | 0.78472838 |
| 2 | Albania | -0.08027103 |
| ---- | ---- | --- |
| 70 | Germany | 0 |
| 71 | Ghana | 0.78954697 |
| 72 | Greece | 0.14346144 |
| 73 | Greenland | 0.66598382 |
| ---- | ---- | ---- |
| 185 | Yemen | 0.8207 |
| 186 | Zambia | 0.8306 |
| 187 | Zimbabwe | 0.8349 |

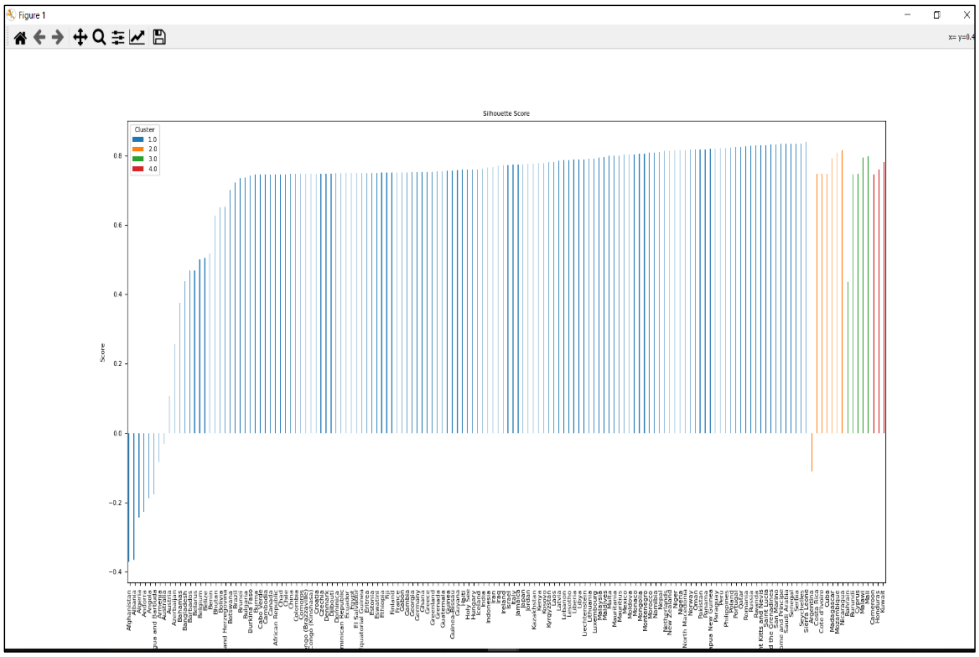


**Figure 4.16 Silhouette Score Graph using Minkowski**

Positive values indicate that the clusters are well separated and distinct. Negative indicates that clusters have been assigned incorrectly. Therefore, the clusters are well apart from each other as the silhouette score is closer to 1. When doing experiment on 1/5/2020, the Euclidean distance with $\epsilon$ = 11,000 and MinPts = 3 is the best performance compared to the others. Based on the above experiments, the average silhouette score of Euclidean distance with $\epsilon$ = 11,000 achieved 0.693, the score of Manhattan distance using $\epsilon$ = 20,000 is 0.684, and the score of Minkowski with $\epsilon$ = 9,000 gave 0.644. Therefore, according to the results, the Euclidean distance with $\epsilon$ = 11,000 and MinPts = 3 is the best performance than the others. In other words, the combination of Minkowski with $\epsilon$ = 9,000 and MinPts = 3 is the worst case as illustrated in Figure 4.17.

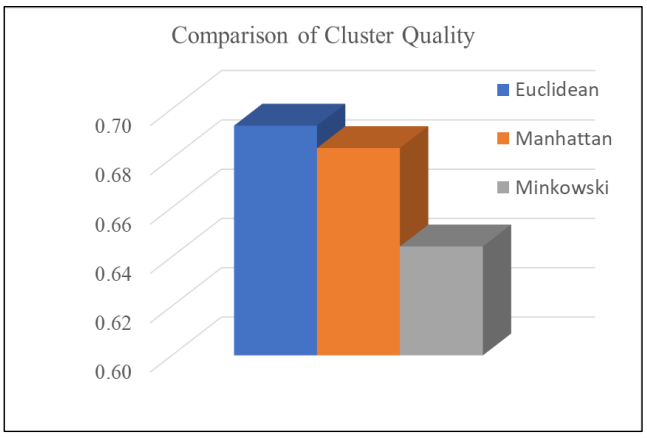


**Figure 4.17 Comparison of Clustering Quality on Three Distance Measures**

## 4.7 Experimental Results based on 16 Trials

Furthermore, author investigate the effectiveness of using distance calculation methods on 16 trials experiment. Then, the sytstem compare and analyze the clustering quality by using average silhouette score. The silhouette score of each trial on using Euclidean, Manhattan and Minkowski methods are described in Table 4.16. Moreover, the graphical view of silhouette score results on each trial is depicted in Figure 4.18.

**Table 4.16 Average Silhouette Score Result for 16 Trials**

| Trials | Start Date | End Date | Euclidean | Manhattan | Minkowski |
|---|---|---|---|---|---|
| 1 | 2/1/2020 | 2/1/2020 | 0.984263 | 0.978507 | 0.980132 |
| 2 | 2/15/2020 | 2/15/2020 | 0.940591 | 0.937009 | 0.934973 |
| 3 | 3/1/2020 | 3/15/2020 | 0.908241 | 0.902293 | 0.906691 |
| 4 | 3/15/2020 | 3/15/2020 | 0.783782 | 0.744113 | 0.783782 |
| 5 | 4/1/2020 | 4/1/2020 | 0.658671 | 0.634313 | 0.649547 |
| 6 | 4/15/2020 | 4/15/2020 | 0.66171 | 0.616712 | 0.616842 |
| 7 | 5/1/2020 | 5/1/2020 | 0.693223 | 0.684119 | 0.644221 |
| 8 | 5/15/2020 | 5/15/2020 | 0.598585 | 0.625801 | 0.596548 |
| 9 | 6/1/2020 | 6/1/2020 | 0.487193 | 0.480176 | 0.486799 |
| 10 | 1/22/2020 | 1/31/2020 | 0.980392 | 0.975413 | 0.974149 |
| 11 | 2/1/2020 | 2/29/2020 | 0.811567 | 0.805707 | 0.799484 |
| 12 | 3/1/2020 | 3/31/2020 | 0.539099 | 0.530717 | 0.531675 |
| 13 | 4/1/2020 | 4/30/2020 | 0.652879 | 0.640005 | 0.558567 |
| 14 | 5/1/2020 | 5/31/2020 | 0.499524 | 0.486427 | 0.488743 |
| 15 | 6/1/2020 | 6/30/2020 | 0.597206 | 0.55853 | 0.547019 |
| 16 | 7/1/2020 | 7/27/2020 | 0.535262 | 0.512605 | 0.511546 |

While doing experiments on 16 trials, using Euclidean distance method achieved average silhouette score of 0.708. With Manhattan distance method, the average silhouette score is 0.694. Using Minkowski distance provided 0.688. Therefore, Euclidean distance method is the best method by analyzing these results.

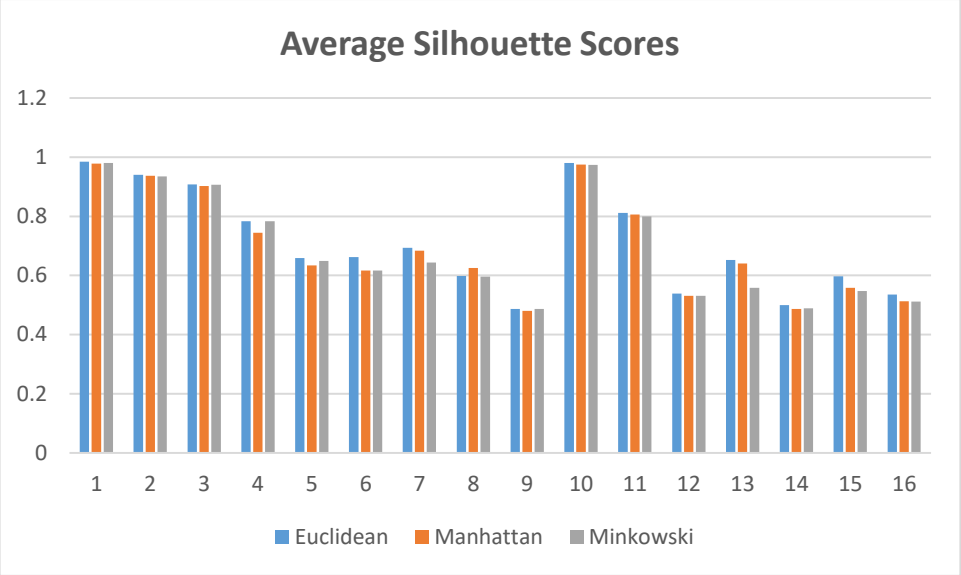


**Figure 4.18 Experimental Result for 16 Trials**

This system limits the minimum number of points (MinPts). The user can select the MinPts from 2 to 14. If the user inputs the wrong MinPts, the system will display the error message as shown in Figure 4.19.

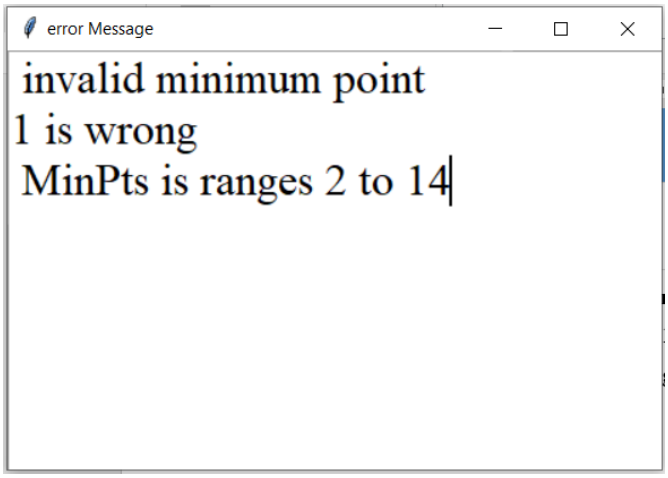


**Figure 4.19 Error Message**

# CHAPTER 5

# CONCLUSION

The job of class identification in spatial datasets is a desirable application for clustering algorithms. However, the application for the big spatial databases raises the following requirements for clustering algorithms: little domain expertise is needed to calculate the input parameters, discovery of clusters with arbitrary shape, and high performance on large databases. The well-known clustering techniques do not offer solution to the combination of these conditions.

This system suggested the clustering algorithm DBSCAN, which relies on a definition of clusters that is density-based and intended to find clusters of arbitrary shape. Only two input parameters are needed for DBSCAN, and it helps the user to choose the right value for each. In a density-based approach, clusters are thought of as areas in the data space where items are concentrated and are separated by areas where there are less noise objects. These regions may be arbitrary shape, and the distribution of the points inside a region may also be arbitrarily distributed. A density-based algorithm called DBSCAN is particularly effective in finding arbitrary-shaped clusters and noise that are present in the database. This information is obtained in advance for each cluster using Epsilon (ϵ) and MinPts. DBSCAN is noticeably more successful in finding clusters of arbitrary shape.

In this system, DBSCAN algorithm is presented for large COVID-19 database. It requires only two parameters (ϵ and MinPts) and supports that the user can choose ϵ value on K-distance Graph. This system is applied to cluster countries that suffer COVID-19 cases. Three distance calculation methods are compared in DBSCAN. According to the clustering results, Euclidean distance achieved the highest clustering performance and Minkowski distance gave the worst performance. By using this system, people can clearly know which country has the similar amount of COVID-19 suffered cases.

## 5.1 Limitation of the System

This system sensitive to two types of clustering parameters MinPts and ϵ. User have to input the MinPts, the range between 2 and 14. If the user inputs the outside number from the range, the system will display error. Moreover, user have to choose

the effective ϵ value based on the k-distance graph. ϵ corresponds to critical change or strong bended area on the curves.

## 5.2 Future Extension

This system has only considered COVID-19 statistical dataset that includes only cases. Therefore, this system can be extended to use other COVID-19 dataset that includes the user suffered symptoms. Investigating DBSCAN applications in high dimensional feature spaces is necessary. Moreover, this system can be tested by using other similarity or distance methods. For analysis, this system can be extended to use other cluster quality analysis methods.

# AUTHOR'S PUBLICATIONS

[1] Min Khant Htway, Hay Mar Soe Naing, "Clustering of Countries based on Number of COVID-19 Cases by using DBSCAN Algorithm", National Journal of Parallel and Soft Computing, University of Computer Studies, Yangon, 2022.

# REFERENCES


[1] O. Godwin and F. N. Ugwokr, "Clustering Algorithm for a Healthcare Dataset using Silhouette Score Value", International Journal of Computer Science & Information Technology (ICSIT), vol. 10, no. 2, pp. 27-37, 2018.

[2] D. J. Divya and S. Prakasha,"Clustering Techniques for Medical Imaging", International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 9, no. 25, pp. 510-516, 2019.

[3] V. Crnogorac, M. Grbic and M. Dukanovic, "Clustering of European Countries and Territories based on Cumulative Relative Number of COVID19 Patients in 2020", IEEE, International Symposium Infoteh-Jahorina, 2021.

[4] A. Doroshenko, "Analysis of the Distribution of COVID-19 in Italy using Clustering Algorithms", IEEE, Third International Conference on Data Stream Mining & Processing, 2020.

[5] R. Pung, C. J. Chiew and B. E. Young, "Investigation of Three Clusters of COVID-19 in Singapore Implications for Surveillance and Response Measures", Elsevier, 2020.

[6] Jiawei Han and Micheline Kamber, “Data Mining Concepts and Techniques”, Simon Fraser University, USA, 2001.

[7] Bing Liu, “Web Data Mining”, Department of Computer Science, University of Illinois at Chicago, USA, 2007.

[8] S. Mukherjee, R. Shaw and N. Haldar, "A Survey of Data Mining Applications and Techniques", International Journal of Computer Science and Information Technologies (IJCSIT), vol. 6, no. 5, pp. 4663-4666, 2015.

[9] P. Kalyani, "Approaches to Partition Medical Data using Clustering Algorithms", International Journal of Computer Applications, vol. 49, no. 23, pp. 7-10, 2012.

[10] K. Kameshwaran and K. Malarvizhi, "Survey on Clustering Techniques in Data Mining", International Journal of Computer Science and Information Technologies, vol. 5, no. 2, pp. 2272-2276, 2014.

[11] E. Martin, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", University of Munich, Germany, 1996.

[12] A. Moreira, M. Y. Santos and S. Carneiro, "Density-based clustering algorithms – DBSCAN and SNN", University of Minho – Portugal, 25. 7. 2005.

[13] M. Ester and H. P. Kriegel, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Institute for Computer Science, University of Munich, Germany.