Annals of Data Science (2024) 11(5):1525–1544
https://doi.org/10.1007/s40745-022-00428-2

An Improved K-means Clustering Algorithm Towards an Efficient Data-Driven Modeling

Md. Zubair[1] · MD. Asif Iqbal[1] · Avijeet Shil[1] · M. J. M. Chowdhury[2] · Mohammad Ali Moni[3] · Iqbal H. Sarker[1]

Received: 6 March 2022 / Revised: 12 May 2022 / Accepted: 27 May 2022 /
Published online: 25 June 2022

Abstract

K-means algorithm is one of the well-known unsupervised machine learning algorithms. The algorithm typically finds out distinct non-overlapping clusters in which each point is assigned to a group. The minimum squared distance technique distributes each point to the nearest clusters or subgroups. One of the K-means algorithm's main concerns is to find out the *initial optimal centroids* of clusters. It is the most *challenging task* to determine the optimum position of the initial clusters' centroids at the very first iteration. This paper proposes an approach to find the optimal initial centroids efficiently to reduce the number of *iterations and execution time*. To analyze the effectiveness of our proposed method, we have utilized different real-world datasets to conduct experiments. We have first analyzed COVID-19 and patient datasets to show our proposed method's efficiency. A synthetic dataset of 10M instances with 8 dimensions is also used to estimate the performance of the proposed algorithm. Experimental results show that our proposed method outperforms traditional kmeans++ and random centroids initialization methods regarding the computation time and the number of iterations.

Keywords K-means Clustering · Principal Component Analysis · Percentile · Unsupervised Algorithm · Machine Learning · Data Science

✉ Iqbal H. Sarker
iqbal@cuet.ac.bd

[1] Department of Computer Science and Engineering, Chittagong University of Engineering & Technology, Chittagong 4349, Bangladesh

[2] Department of Computer Science and Information Technology, La Trobe University, Victoria 3086, Australia

[3] School of Health and Rehabilitation Sciences, Faculty of Health and Behavioural Sciences, The University of Queensland, St Lucia QLD 4072, Australia

1 Introduction

Machine learning is a subset of Artificial Intelligence that makes applications capable of learning and improves the result through the experience, not being programmed explicitly through computation [1]. Supervised and unsupervised are the basic approaches of the machine learning algorithm. The unsupervised algorithm identifies hidden data structures from unlabelled data contained in the dataset [2]. According to the hidden structures of the datasets, clustering algorithm is typically used to find the similar data groups which also can be considered as a core part of data science as mentioned in Sarker et al. [3].

In the context of data science and machine learning, K-Means clustering is known as one of the powerful unsupervised techniques to identify the structure of a given dataset. The clustering algorithm is the best choice for separating the data into groups and is extensively exercised for its simplicity [4]. Its applications have been seen in different important real-world scenarios, for example, recommendation systems, various smart city services as well as cybersecurity and many more to cluster the data. Beyond this, clustering is one of the most useful techniques for business data analysis [5]. Also, the K-means algorithm has been used to analyze the users' behavior and context-aware services [6]. Moreover, the K-means algorithm plays a vital role in complicated feature extraction.

In terms of problem type, K-means algorithm is considered an NP-Hard problem [7]. It is widely used to find the number of clusters so that it is possible to divide the unlabelled dataset into clusters to solve real-world problems in various application domains, mentioned above. It is done by calculating the distances from a centroid of a cluster. We need to fix the initial centroids' coordinates to find the number of clusters at initialization. Thus, this step has a crucial role in the K-means algorithm. Generally, we randomly select the initial centroids. If we can determine the initial centroids efficiently, it will take fewer steps to converge. According to D. T. Pham et al. [8], the overall complexity of the K-means algorithm is

$$O(k^2 * n * t) \tag{1}$$

Where t is the number of iterations, k is the number of clusters and n is the number of data points.

Optimization plays an important role both for supervised and unsupervised learning algorithms [9]. So, it will be a great advantage if we can save some computational costs by optimization. The paper will give an overview to find the initial centroids more efficiently with the help of principal component analysis (PCA) and percentile concept for estimating the initial centroids. We need to have fewer iterations and execution times than the conventional method.

In this paper, recent datasets of COVID-19, a healthcare dataset and a synthetic dataset size of 10 Million instances are used to analyze our proposed method. In the COVID-19 dataset, K-means clustering is used to divide the countries into different clusters based on the health care quality. A patient dataset with relatively high instances and low dimensions is used for clustering the patient and inspecting the performance of the proposed algorithm. And finally, a high instance of 10M synthetic dataset is used to evaluate the performance. We have also compared our method with the kmeans++ and random centroids selection methods.

The key contributions of our work are as follows-

- We propose an improved K-means clustering algorithm that can be used to build an efficient data-driven model.
- Our approach finds the optimal initial centroids efficiently to reduce the number of iterations and execution time.
- To show the efficiency of our model comparing with the existing approaches, we conduct experimental analysis utilizing a COVID-19 real-world dataset. A 10M synthetic dataset as well as a health care dataset have also been analyzed to determine our proposed method's efficiency compared to the benchmark models.

This paper provides an algorithmic overview of our proposed method to develop an efficient k-means clustering algorithm. It is an extended and refined version of the paper [10]. Elaborations from the previous paper are (i) analysis of the proposed algorithm with two additional datasets along with the COVID-19 dataset [10], (ii) the comparison with random centroid selection and kmeans++ centroid selection methods, (iii) analysis of our proposed method more generalized way in different fields, and (iv) including more recent related works and summarizing several real-world applications.

In Sect. 2, we'll discuss the works of similar concepts. In Sect. 3, we will further discuss our proposed methodology with a proper example. In Sect. 4, we will show some experimental results, description of the datasets and a comparative analysis. In Sects. 5 and 6, discussion and conclusion have been included.

2 Related Work

Several approaches have been made to find the initial cluster centroids more efficiently. In this section, we will bring some of these works. M. S. Rahman et al. [11], provided a centroids selection method based on radial and angular coordinates. The authors showed experimental evaluations for his proposed work for small(10k-20k) and large(1M-2M) datasets. However, the number of iterations of his proposed method isn't constant for all the test cases. Thus, the runtime of his proposed method increases drastically with the increment of the cluster number. A. Kumar et al. also proposed to find initial centroids based on the dissimilarity tree [12]. This method improves k-means clustering slightly, but the execution time isn't significantly enhanced. In [13], M.S. Mahmud et al. proposed a novel weighted average approach to finding the initial centroids by calculating the mean of every data point's distance. It only describes the execution time of 3 clusters with 3 datasets. Improvement of execution time is also trivial. In [14], authors M. Goyal et al. also tried to find the centroids by dividing the sorted distances with k, the number of equal partitions. This method's execution time has not been demonstrated. M. A. Lakshmi et al. [15] proposed a method to find initial centroids with the help of the nearest neighbour method. They compared their method using SSE(Sum of the Squared Differences)with random and kmeans++ initial centroids selection methods. SSE of their method was roughly similar to random and kmeans++ initial centroids selection methods. Moreover, they didn't provide any comparison regarding execution time as well. K. B. Sawant [16] has proposed a method to find the initial cluster with the distance of the neighbourhood. The proposed method calculates and sorts all the distances from the first point. Then, the entire dataset was divided into equal portions. But the author didn't mention any comparative analysis to prove his proposed method better than the existing one. In [17] , the authors proposed to save the distance to the nearest cluster of the previous iteration and used the distance to compare in the next iteration. But still, initial centroids are selected randomly. In [18], M. Motwani et al. proposed a method with the farthest distributed centroids clustering (FDCC) algorithm. The authors failed to include information on how this approach performs with a distributed dataset and a comparison of execution times. M. Yedla et al. [19] proposed a method where the algorithm sorted the data point by the distance from the origin and subdivided it into k (number of clusters needed) sets.

COVID-19 has been the subject of several major studies lately. The K-means algorithm has a significant influence on these studies. S. R. Vadyala et al. proposed a combined algorithm with k-means and LSTM to predict the number of confirmed cases of COVID-19 [20]. In [21], author A. Poompaavai et al. attempted to identify the affected areas by COVID-19 in India by using the k-means clustering algorithm. Many approaches have been attempted to solve the COVID-19 problem using k-means clustering. In [22], S.K. Sonbhadra et al. proposed a novel bottom-up approach for COVID-19 articles using k-means clustering along with DBSCAN and HAC. S. Chinchorkar used the K-means algorithm for defining Covid-19 containment zones. In this paper [23], the size and locations of such zones (affected by Coronapositive patients) are considered dynamic. K-means algorithm is proposed to handle the zones dynamically. But if the number of Corona-positive patient outbreaks, K-means may not be effective as it will take huge computational power and resources to handle the dataset. N. Aydin et al. used K-means in accessing countries' performance against COVID-19. In this paper [24], K-means and hierarchical clustering methods are used for cluster analysis to find out the optimum number of classes in order to categorize the countries for further performance analysis. In this paper [25], a comparison is made based on the GDP declines and deaths in China and OECD countries. K-means is used for clustering analysis to find out the current impact of GDP growth rate, deaths, and account balances.T. Zhang used a generalized K-means algorithm in GLMs to group the state-level time series patterns for the analysis of the outbreak of COVID-19 in the United States [26]

K-means clustering algorithm has a huge impact on patient and medically related work. Many researchers use the k-means algorithm for their research purpose. Ldl Fuente-Tomas et al. [27] used the k-means algorithm to classify patients with bipolar disorder. P. Sili-tonga et al. [28] used a k-means clustering algorithm for clustering patient disease data. N. Das et al. [29] used the k-means algorithm to find the nearest blood & plasma donor. MS Alam et al. [30] used the k-means algorithm m for detecting human brain tumors in a magnetic resonance imaging (MRI) image. Optimized data mining and clustering models can provide an insightful information about the transmission pattern of COVID-19 outbreak [31].

Among all the improved and efficient k-means clustering algorithms proposed previously, they take the initial center by randomization [15, 17] or the k-means++ algorithm [15, 32]. Those processes of selecting the initial cluster take more time.

In contrast, our proposed k-means algorithm chooses initial centroids using principal component analysis(PCA) & percentile.

3 Proposed Methodology

The K-means clustering is the NP-Hard optimization problem [33]. The efficiency of the k-means clustering algorithm depends on the selection or assignment of initial clusters' centroids [34]. So, it is important to select the centroids more systematically to improve the K-means clustering algorithm's performance and execution time. This section introduces our proposed method of assignment of initial centroids by using Principal Component Analysis (PCA) and dividing the values into percentiles to get the efficient initial coordinate of centroids. The flowchart in Fig. 1 depicts the overall process of our proposed method.

In the next subsection, We will describe our proposed method.

3.1 Input Dataset and Pre-processing

In Sect. 4.1, a vivid description has been given about the dataset of our model. Every dataset has properties that requires manual pre-processing to make it competent for feeding the K-means clustering algorithm. K-means clustering algorithm can only perform its operation on numerical data. So any value which is not numerical or categorical must need to be transformed. Again, handling missing values needs to be performed during manual pre-processing.

As we have tried to solve real-world problems with our proposed method, all attributes are not equally important. So, we have selected some of the attributes for the K-means clustering algorithm's implementation. So, choosing the right attributes must be specified before applying Principal Component Analysis (PCA)and percentile.

3.2 Principal Component Analysis (PCA)

PCA is a method, mostly discussed in mathematics, that utilizes an orthogonal transformation to translate a set of observations of potentially correlated variables into a set of values of linearly uncorrelated variables, called principal components. PCA is widely used in data analysis and making a predictive model [35]. PCA reduces the dimension of datasets by increasing interpretability but minimizing the loss of information simultaneously. For this purpose, the orthogonal transformation is used. Thus, the PCA algorithm helps to quantify the relationship between the large related dataset [36], and it helps to reduce computational complexity. A pseudo-code of PCA algorithm is provided below 1.

PCA tries to fit as much information as possible into the first component, then the second component, and so on.

We convert the multi-dimensional dataset into two dimensions for our proposed method using PCA. Because with these two dimensions, we can easily split the data into horizontal and vertical planes.

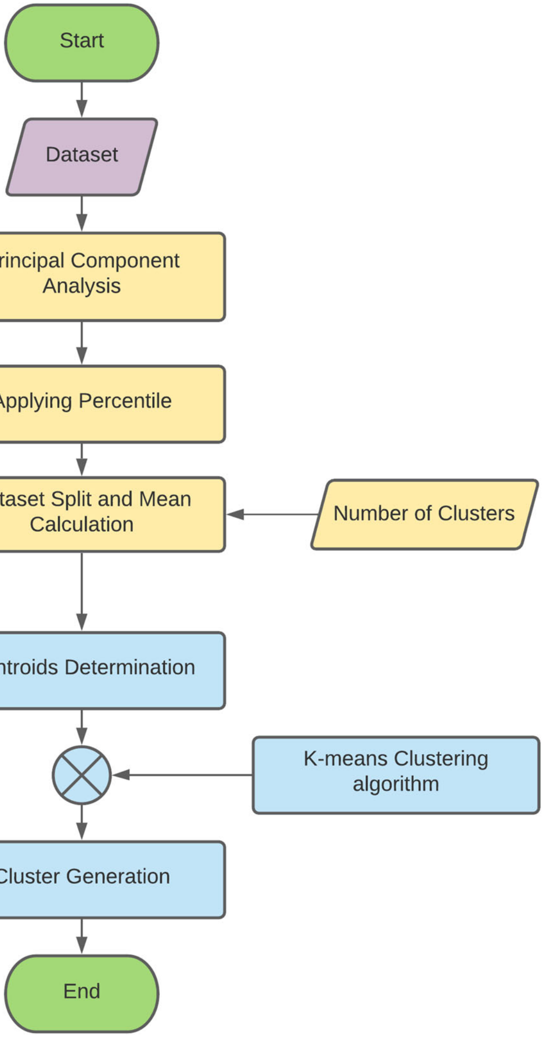


Fig. 1 Flowchart of the proposed method

3.3 Percentile

The percentile model is a well-known method used in statistics. It divides the whole dataset into 100 different parts. Each part contains 1 percent data of the total dataset. For example, the 25th percentile means this part contains 25 percent of data of the total dataset. That implies, using the percentile method, we can split our dataset into different distributions according to our given values [37].

The percentile formula is given below:

$$R = \frac{P}{100} * (n + 1)$$

Algorithm 1 PCA Algorithm

```
Compute the mean, $\mu = \frac{1}{p}\sum_{k=1}^{p} x_k$, where p = Number of Pattern, $x_k$ = Feature Matrix, and k is an iterator.
Compute covariance matrix, $C = \frac{1}{p}\sum_{k=1}^{p}\{x_k - \mu\}\{x_k - \mu\}^T$, where T represents the Transpose matrix.
Compute Eigenvalues $\tau_i$ and Eigenvectors $V_i$,
    $C_i = \tau_i V_i$
    where i = (1,2,3…. Number of feature)
Estimate the high-valued Eigenvectors.
Extract low dimensional feature vectors (principal components or PC) from raw feature matrix.
```

Here, P = The Percentile to find, n = Total Number of values, R = Percentile at P

After reducing the dataset into two dimensions by applying PCA, the percentile method is used on the dataset. The first component of PCA holds the majority of information, and Percentiles can only be applied to one-dimensional data. As we have found most of the information in the first component of PCA, we have considered only the first component.

Finally, dataset is partitioned according to the desired number of clusters with the help of the percentile method. It splits the two-dimensional data into equal parts of the desired number of clusters.

3.4 Dataset Split and Mean Calculation

After splitting the reduced dimensional dataset through percentile, we extract the split data from the primary dataset by indexing for every percentile. In this process, we can get back the original data. After retrieving the original data for each percentile, we have calculated the mean of each attribute. These means from the split dataset are the initial centroids of clusters of our proposed method.

3.5 Centroids Determination

After splitting the dataset and calculating the mean according to Subsect. 3.4, we select each split dataset's mean as a centroid. These centroids are considered as our proposed initial centroids for the efficient k-means clustering algorithm. K-means is an iterative method that attempts to make partitions in an unsupervised dataset. The sub-groups formed after the division are non-overlapping subgroups. This algorithm tries to make a group as identical as possible for the inter-cluster data points and aims to remain separate from the other cluster. In Algorithm 2, a pseudo-code is provided of the k-means algorithm:

Algorithm 2 K-means Clustering Algorithm

```
Input: A dataset D and Number of clusters K
Output: K number of clusters
procedure K- MEANS($Input$)
    Take initial centroids C= c1,c2,c3…..,ck
    Assign each instance d1 to a cluster, Ki by the closest distance
    Calculate the new centroids by taking mean of each cluster
    Repeat the above process until the centroids converges
```

3.6 Cluster Generation

At the last step, we have executed our modified k-means algorithm until the centroids converge. Passing our proposed centroids instead of random or kmeans++ centroids through the k-means algorithm we have generated the final clusters [32]. The proposed method always considers the same centroids for each test. The pseudocode of our whole proposed methodology is given in the algorithm 3. In the next section, evaluation and experimental results of our proposed model are discussed.

Algorithm 3 Proposed Method of Initial Cluster Centroids

```
Input: A dataset **D** and Number of clusters **K**
Output: Efficient initial centroids for **K** clusters
procedure PROPOSED METHOD($Input$)
    All n attributes $a1, a2, a3, \dots, an$ of **D** must be numeric. If there is any non-numeric attribute, just convert it to numeric value.
    Apply Principal Component Analysis (PCA) with 2 components to the dataset, **D**.
    Apply percentile for splitting the whole dataset into **K** equal parts based on 1st component.
    Extract the split dataset from primary data by index.
    Calculate the mean of each attribute of the split datasets.
    Take the mean of each dataset as the initial clusters centroids, $C = c1, c2, \dots, ck$, where $c1, c2, .., ck$ are the initial centroids for 1st,2nd,….,k clusters consecutively.
    Assign the centroids to the k-means clustering algorithm
```

4 Evaluation and Experimental Result

We have gone through a couple of experiments to measure the effectiveness and validate our proposed model for selecting the optimum initial centroids for the k-means clustering algorithm. The proposed model is tested with the high dimension, relatively high instances, and very high instances datasets. We have used a few COVID-19 datasets and merged them to have a handful of features for clustering the countries according to their health quality during COVID-19. We have tested the model for clustering the health care patients [38]. The mode is also tested with 10 million data created with the scikit-learn library [39]. A detailed explanation of the datasets is given in the following subsection.

Table 1 Sample data of COVID-19 dataset

Country	Total cases per million	New cases per million	Total deaths per million	New deaths per million	Cardiovasc death rate	Hospital beds per thousand	life expectancy
Australia	839.102	12.275	12.275	0.706	107.791	3.84	83.44
Bangladesh	1581.808	17.651	20.876	0.237	298.003	0.8	72.59
China	61.769	0.079	3.258	0.003	261.899	4.34	76.91

Table 2 Sample data of Covid-19-testing-policy

Entity	Code	Date	Testing policy
Australia	AUS	Aug 11, 2020	3
Bangladesh	BGD	Aug 11, 2020	2
China	CHN	Aug 11, 2020	3

4.1 Dataset Exploration

We experimented with many different datasets to descry our model's efficiency. Properties of our used datasets are

- Low instances with a high dimensional dataset
- Relatively high instances with a low dimensional dataset
- Very high instances dataset

In the next Subsect. of 4.1.1, 4.1.2 and 4.1.3, a brief explanation of those datasets is given.

4.1.1 COVID-19 Dataset

Many machine learning algorithms, including supervised and unsupervised methods, were applied to the covid-19 dataset. For creating our model, we used a few datasets for selecting the features required for analyzing the health care quality of the countries. The selected datasets are owid-covid-data [40], covid-19-testing-policy [41], public-events-covid [41], covid-containment-and-health-index [41], inform-covid-indicators [42]. It is worth mentioning that we used the data up to 11th August 2020.

For instance, some of the attributes of the owid-covid-data [40] are shown in Table 1. Covid-19-testing-policy [41] dataset contains the categorical values of the testing policy of the countries shown in Table 2.

Other datasets also contained such features required to ensure the health care quality of a country. These real-world datasets helped us to analyze our proposed method for real-world scenarios.

We merged the datasets according to the country name with the regular expression pre-processing. Some pre-processing and data cleaning had been conducted in case of merging the data, and we had also handled some missing data consciously. There are so many attributes regarding COVID-19; among them, 25 attributes were finally selected, as these attributes closely signify the health care quality of a country. The

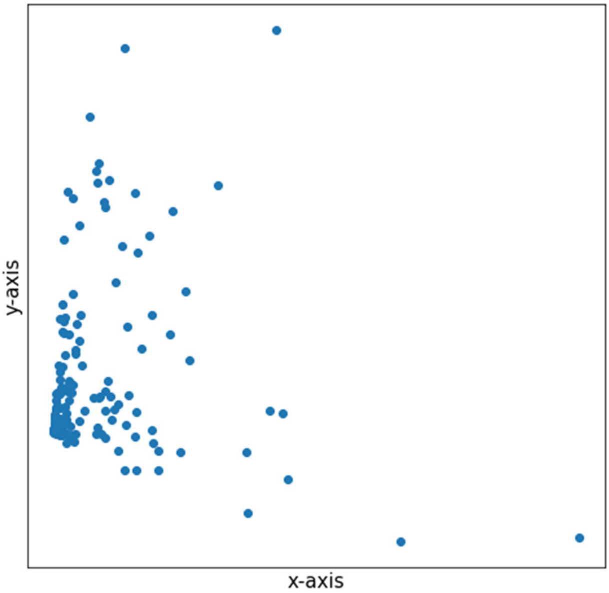


Fig. 2 2D distribution plot of COVID-19 dataset

attributes represent categorical and numerical values. These are country name, cancellation of public events (due to public health awareness), stringency index[1], testing policy (category of testing facility available to the mass people), total positive case per million, new cases per million, total death per million, new deaths per million, cardiovascular death rate, hospital beds available per thousand, life expectancy, inform the COVID-19 risk (rate), hazard and exposure dimension rate, people using at least basic sanitation services (rate), inform vulnerability(rate), inform health conditions (rate), inform epidemic vulnerability (rate), mortality rate, prevalence of undernourishment, lack of coping capacity, access to healthcare, physicians density, current health expenditure per capita, maternal mortality ratio. We have consciously selected the features before feeding the model. A two-dimensional plot of the dataset is shown in Fig. 2.

It is a dataset of low instances with high dimensions.

4.1.2 Medical Dataset

Our second dataset is Medical related dataset with 100k instances. It is an open-source dataset for research purposes. It has a random sample of 6 million patient records from our Medical Quality Improvement Consortium (MQIC) database [38]. Any personal information has been excluded. The final attributes of the dataset are: gender, age, diabetes, hypertension, stroke, heart disease, smoking history, and BMI.

[1] It is one of the matrices used by Oxford COVID-19 Government Response Tracker [43]. It delivers a picture of the country's enforced strongest measures.

Table 3 Sample data of Medical Dataset

Gender	Age	Diabetes	Hypertension	Stroke	Heart disease	Moking history	BMI
Female	80.0	0	0	0	1	Never	25.19
Female	36.0	0	0	0	0	Current	23.45
Female	44.0	1	0	0	0	Never	19.31
Male	42.0	0	0	0	0	Never	33.64
Male	18.0	0	0	0	0	Never	21.78

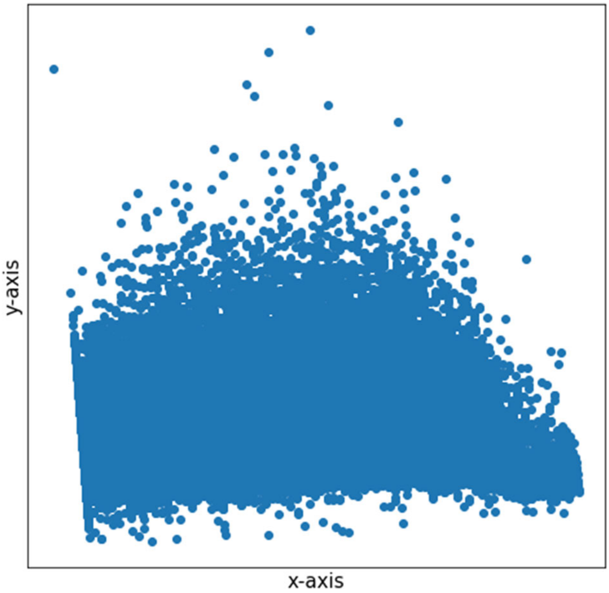


Fig. 3 2D distribution plot of medical dataset

A glimpse of the Medical dataset is given in Table 3. It is a dataset of relatively high instances with low dimensional data. Figure 3 provides a graphical representation of the dataset, which gives meaningful insights into the distribution of the data.

4.1.3 Synthetic Dataset

A final syntactic dataset has been made to cover the very high instances dataset category. This synthetic dataset is created with the scikit-learn library [39]. The dataset has 8 dimensions with 10M (Ten Million) instances. The two-dimensional distribution is shown in Fig. 4.

4.2 Experimental Setup

We have selected the following questions to evaluate our proposed model.

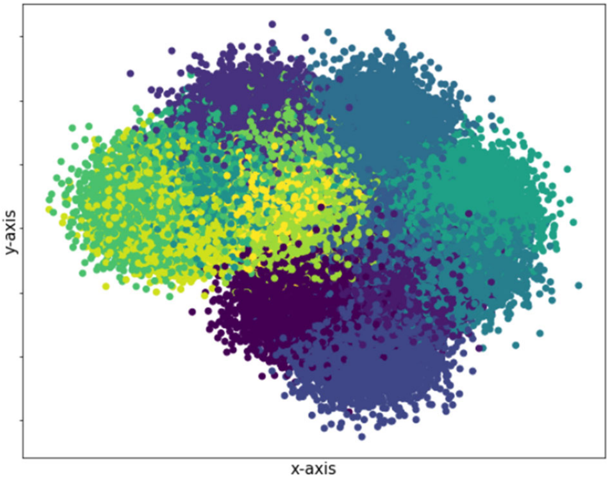


Fig. 4 2D distribution plot of scikit-learn library dataset

- Is the proposed method for selecting efficient clusters of centroids working well both for the high and low dimensional dataset?
- Does the method reduce the iteration for finding the final clusters with the k-means algorithm compared to the existing methods?
- Does the method reduce the execution time to some extent compared to the existing methods for the k-means clustering algorithm?

To answer these questions, we have used real-time COVID-19 healthcare quality data of different countries, patient data, and a scikit-learn library dataset with 10M instances. In the following sub-sections, we have briefly discussed experimental results and comparison with its' effectiveness.

4.3 Evaluation Process

In machine learning and data science, computational power is one of the main issues. Because at the same time, the computer needs to process a large amount of data. So, reducing computational costs is a big deal. K-means clustering is a popular unsupervised machine learning algorithm. It is widely used in different clustering processes. The algorithm randomly selects the initial clusters' centroids, which sometimes causes many iterations and high computational power. As discussed in the methodology section, we have implemented our proposed method. However, many researchers proposed many ideas discussed in the related work Sect. 2. We have compared our method with the best existing k-means++ and random method [32]. We have measured the effectiveness of the model with

– Number of iterations needed for finding the final clusters
– Execution time for reaching out to the final clusters

These two things will be measured in the upcoming subsections.

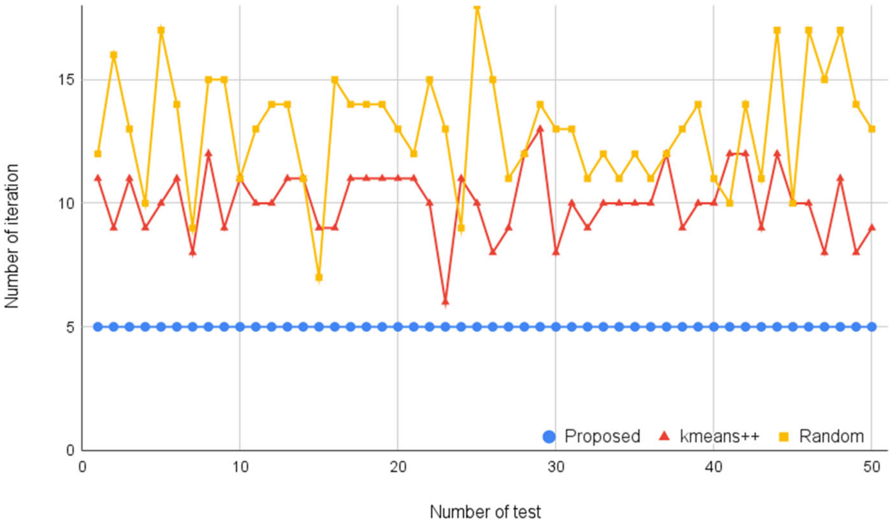


Fig. 5 Iteration for 4 clusters with COVID-19 dataset

4.4 Experimental Result

Our main goal is to improve the traditional k-means clustering algorithm's performance. In this sub-section, we will show the experimental results with different datasets. We have used high dimension datasets, large instances with relatively low dimensions and very high instances. So, that will cover most of the real-world scenarios. Before jumping to the demonstration, it is worth mentioning that we have conducted the experiment on Intel® Core™ i7-8750H processor.

4.4.1 Analysis with COVID-19 Dataset

Firstly, we are starting our experiment with the COVID-19 dataset. Details explanation of the dataset is provided in Subsect. 4.1.1. As we are making clusters for the countries with similar types of health care quality, we have defined the optimum number of clusters with the elbow method [44]. For the COVID-19dataset, the optimum number of clusters is 4. So, we are looking forward to the results with 4 clusters. Here, each type of cluster contains the same types of healthcare quality.

In Fig. 5, we have shown the experimental result in terms of iteration number for the COVID-19 dataset of 50 tests. Here, we have compared the results of our proposed method to the traditional random centroid selection method and existing the best centroid selection method kmeans++. The yellow, red, and blue lines in Fig. 5 represent the random, kmeans++, and our proposed method consecutively. The graphical representation clearly shows that the number of iterations of our model is constant and outperforms most of the cases. On the other hand, the random and kmeans++ method propagates randomly, and in most cases, the number of iterations is higher than our proposed method. So, we can claim that our model outperforms in terms of iteration number and execution time.

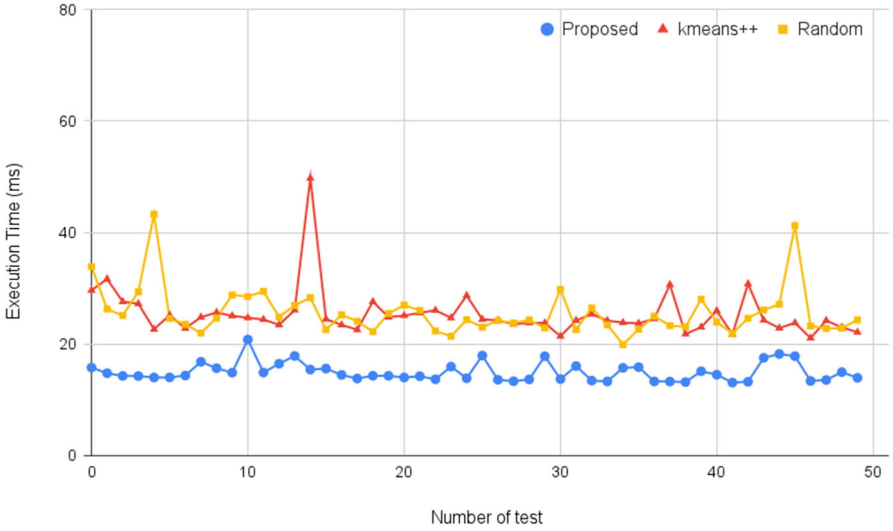


Fig. 6 Execution time for 4 clusters with COVID-19 dataset

The graph in Fig. 6 represents the experimental result for execution time with the COVID-19 data. Execution time is closely related to the number of iterations. In Fig. 6, the yellow line represents the results for the random method, the red line for the existing kmeans++method and the blue line for our proposed method. As the centroids of our proposed method are constant for each iteration, the execution time is always nearly constant. Figure 6 depicts that in the case of the random and kmeans++ method, the execution time varies randomly while our proposed methods outperform in every test case.

4.4.2 Analysis with Medical Dataset

K-means clustering algorithm is widely used in medical sectors as well. Patient clustering is also one of the important tasks where clustering is usually used. We have analyzed our model with the patient data mentioned in Subsect. 4.1.2. As it is a real-world implementation, we have used the elbow method to find out the clusters' optimum number. For the dataset, the optimum number of clusters is 3

We have conducted 50 tests over the dataset. The final outcome in terms of iteration is shown in Fig. 7. The blue line represents the execution time for 3 clusters with the dataset, and it is nearly constant. Compared to the other two methods represented with the yellow line for random and red line for kmeans++, our model notably outperforms.

In Fig. 8, we have shown the experimental result for execution time. The blue line represents the execution time for 3 clusters with the dataset. It is nearly constant. Compared to the other two methods represented with the yellow line for random and red line for kmeans++, our model notably outperforms.

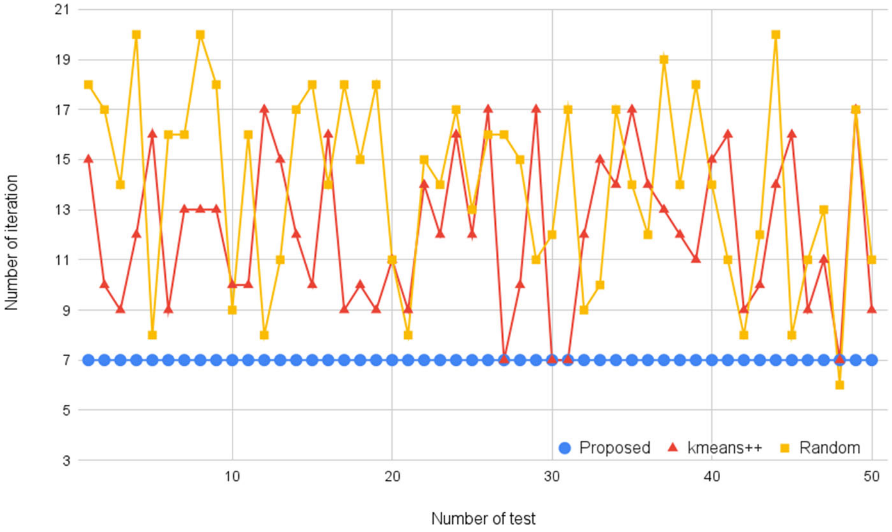


Fig. 7 Iteration for 3 clusters with Medical Dataset

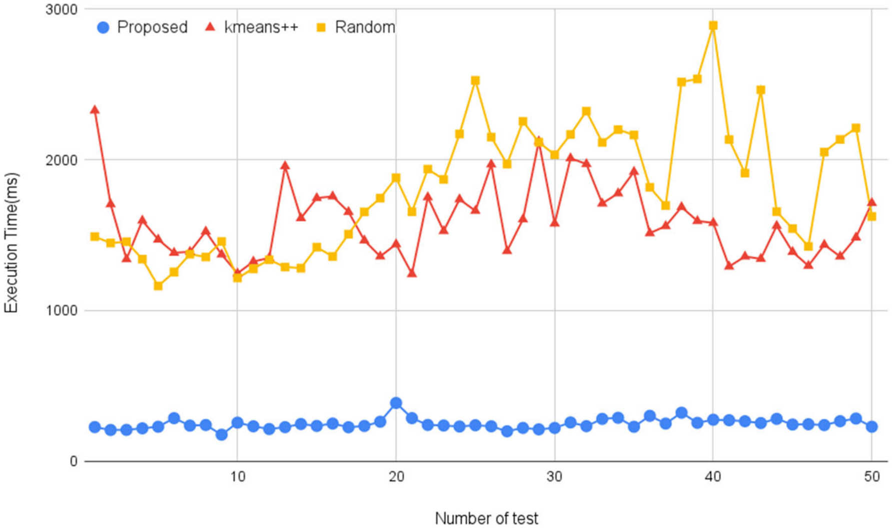
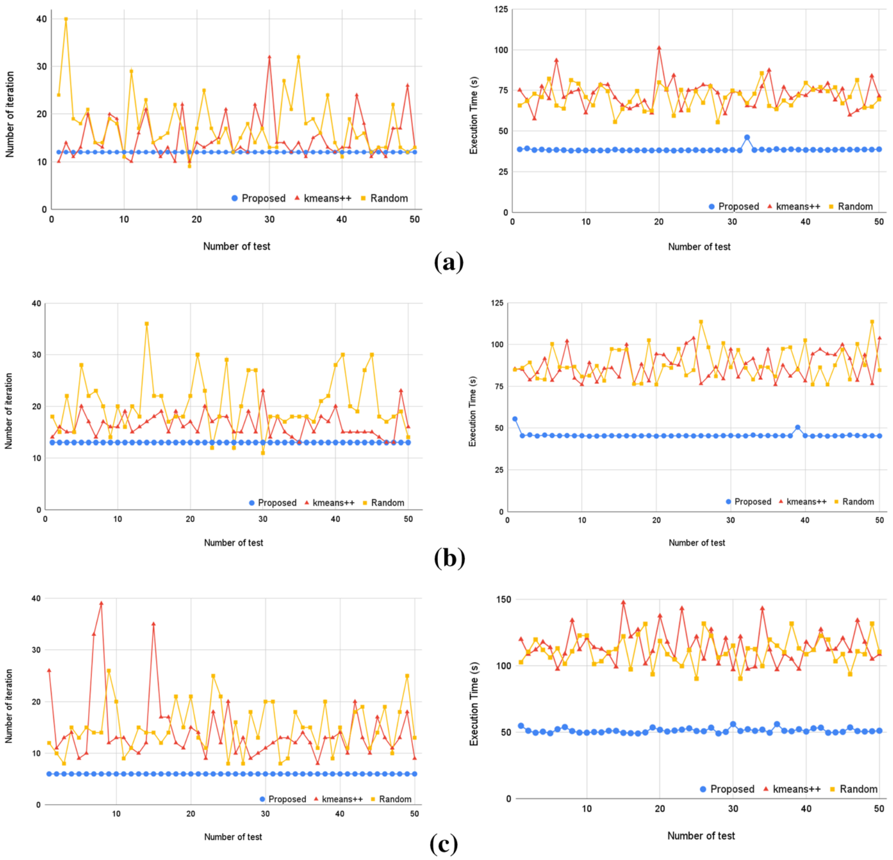


Fig. 8 Execution time for 3 clusters with Medical Dataset

4.4.3 Analysis with Synthetic Dataset

In practical scenarios, clustering data may be massive. For that reason, we have created a synthetic dataset described in Subsect. 4.1.3 for getting insight into our model, whether it works for a very large dataset or not. This synthetic dataset contains about 10 million instances, and the dataset is only created for demonstration purposes. We have also created the clusters randomly for testing our model. We have run the model



(a)

(b)

(c)

Fig. 9 Total iteration and execution time for different number of clusters with Synthetic Dataset. **(a)** 3 clusters **(b)** 4 clusters **(c)** 5 clusters

with random, kmeans++ and our proposed model 50 times simultaneously with the dataset. The model is tested with 3, 4 and 5 clusters. Figure 9 graphically represents the experimental results with the synthetic dataset. The blue, yellow and red lines represent the results for the proposed, random and kmeans++ methods consecutively. The left side graphs show the experimental results in terms of iteration, and the right side graphs show the experimental results for the execution time of 3, 4, and 5 clusters.

For the clusters, the number of iterations is constant for our proposed method, and it also outperforms most of the test cases. Other models are random in nature. The kmeans++ and random models have not reduced the iteration significantly. It is a remarkable contribution.

Execution time is a vital issue for improving an algorithm's performance. The right side graphs of Fig. 9 show the results in terms of execution time for 3, 4, and 5 clusters. The execution time needed for our proposed method is nearly constant and outperforms compared to the kmeans++ and random methods.

4.5 Effectiveness Comparison Based on Number of Iterations and Execution Time

Computation cost is one of the fundamental issues in data science. If we can reduce the number of iterations to some extent, the performance will be improved. A Constant iteration of the algorithm helps us to have the same performance over time.

In Fig. 5, we find the experimental result for covid-19 dataset. When our proposed method is applied, we have found a constant number of iterations in each test. But the existing kmeans++ and random methods' iteration are random. It varies for different test cases. Figure 6 provides the execution time comparison for existing kmeans++, random and our proposed method. Our model executed the k-means clustering algorithm for each test case in the shortest time.

The medical dataset contains a relatively high instance of data with real-world hospital patient data. Figures 7 and 8 represent the experimental results for random, kmeans++ and our proposed methods. We have found that our model converged to the final clusters with a reduced number of constant iterations and improved execution time.

In Fig. 9, we have shown the experimental results of the K-means clustering algorithm for 3,4 and 5 clusters consecutively for the number of iterations along with execution time. This experiment has been done on a synthetic dataset described in Subsect. 4.1.3. Constant optimum iteration compared to the existing model kmeans++, random and shortest execution time signify that our model outperforms for large datasets.

Based on the experimental results, we claim that our model outperforms in real-world applications and reduces the computational power of the K-means clustering algorithm. It also outperforms for a huge instance of datasets.

The above discussion answers the last two questions discussed in the experimental setup Subsect. 4.2.

5 Discussion

K-means clustering is one of the most popular unsupervised clustering algorithms. By default k-means clustering algorithm randomly selects the initial centroids. Sometimes, it consumes a huge computational power and time. Over the years, many researchers tried to select the initial centroids more systematically. Some of the works have been mentioned in Sect. 2, and most of the previous works are not detailed enough and well recognized. From equation 1, we presume that the execution time will be reduced significantly if we can reduce the number of iteration. And our proposed method focuses on minimizing the iteration. Two statistical models, PCA(principal component analysis) [35] and percentile [37] are used to deploy the proposed method. It is also mentionable that the proposed method provides the optimal number of iterations for applying the K-means clustering algorithm. However, the most popular and standard method is kmeans++ [32]. So, we have compared our model with the kmeans++ method and the default random method to analyze the efficiency. We haven't modified the main K-means clustering algorithm instead developed an efficient centroid selection process that provides an optimum number of constant iterations. As the time complexity is directly related to the iteration shown in equation 1 and our model gives less number of iterations, the overall execution time is reduced.

We have made clusters of the countries with similar types of health care quality for solving the real-world problem with our proposed method. It is high-dimensional data. The medical-related dataset is also used for making patient clusters. We have also used a synthetic dataset created with scikit-learn consisting of 10M instances to ensure that our model is also outperforming for a large number of instances. This model performs well for both low and high-dimensional datasets. Thus, this technique could be applied to solve many unsupervised learning problems in various real-world application areas ranging from personalized services to today's various smart city services and security, i.e., to detect cyber-anomalies [6, 45]. Internet of Things (IoT) is another cutting edge technology where clustering is widely used [46].

Our proposed method reduces the computational power. So, the proposed model will work faster in case of a clustering problem where the data volume is too large. This proposed method is easy to implement, and no extra setup is needed.

6 Conclusion

In this article, we have proposed an improved K-means clustering method that increases the performance of the traditional one. We have significantly reduced the number of iterations by systematically selecting the initial centroids for generating the clusters. PCA and Percentile techniques have been used to reduce the dimension of data and segregate the dataset according to the number of clusters. Finally, these segregated data have been used to select our initials centroids. Thus, we have successfully minimized the number of iterations. As the complexity of the traditional K-means clustering algorithm is directly related to the number of iterations, our proposed approach outperformed compared to the existing methods. We believe this method could play a significant role for data-driven solutions in various real-world application domains.

Author Contributions All authors equally contributed to preparing and revising the manuscript.

Funding Not Applicable

Data Availability Statement Data and codes used in this work can be made available upon reasonable request

Declarations

Conflict of Interests The authors declare no conflict of interest.

Ethical statements The authors follow all the relevant ethical rules.

References

1. Sarker IH (2022) Ai-based modeling: Techniques, applications and research issues towards automation, intelligent and smart systems. SN Computer Science 3(2):1–20
2. Bonaccorso G (2017) Machine learning algorithms

3. Sarker IH (2021) Data science and analytics: an overview from data-driven smart computing, decision-making and applications perspective. SN Computer Science 2(5):1–22
4. Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques
5. Olson DL, Shi Y, Shi Y (2007) Introduction to business data mining, vol 10. McGraw-Hill/Irwin, New York
6. Sarker IH, Colman A, Han J, Watters PA (2021) Context-aware machine learning and mobile data analytics: automated rule-based services with intelligent decision-making. Springer Nature, Switzerland
7. Vattani A (2009) The hardness of k-means clustering in the plane. Manuscript, accessible at http://cseweb.ucsd.edu/avattani/papers/kmeans_hardness.pdf, 617
8. Pham DT, Dimov SS, Nguyen CD (2004) An incremental k-means algorithm. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 218(7):783–795
9. Shi Y, Tian Y, Kou G, Peng Y, Li J (2011) Optimization based data mining: theory and applications. Springer, London
10. Zubair Md, Iqbal A, Shil A, Haque E, Moshiul Hoque M, Sarker IH (2020) An efficient k-means clustering algorithm for analysing covid-19. In International Conference on Hybrid Intelligent Systems, pages 422–432. Springer
11. Rahim MdS, Ahmed T (2017) An initial centroid selection method based on radial and angular coordinates for k-means algorithm. In 2017 20th International Conference of Computer and Information Technology (ICCIT), 1–6. IEEE
12. Kumar A, Gupta SC (2015) A new initial centroid finding method based on dissimilarity tree for k-means algorithm. arXiv preprint arXiv:1509.03200
13. Mahmud MdS, Rahman MdM, Akhtar MdN (2012) Improvement of k-means clustering algorithm with better initial centroids based on weighted average. In 2012 7th International Conference on Electrical and Computer Engineering, 647–650. IEEE
14. Goyal M, Kumar S (2014) Improving the initial centroids of k-means clustering algorithm to generalize its applicability. Journal of The Institution of Engineers (India): Series B 95(4):345–350
15. Lakshmi MA, Daniel GV, Rao DS (2019) Initial centroids for k-means using nearest neighbors and feature means. In Wang J, Reddy GRM, Prasad VK, Reddy VS (eds), Soft Computing and Signal Processing, 27–34, Singapore. Springer Singapore
16. Sawant KB (2015) Efficient determination of clusters in k-mean algorithm using neighborhood distance. The International Journal of Emerging Engineering Research and Technology 3(1):22–27
17. Fahim AM, Salem AM, Torkey FAf, Ramadan MA (2006) An efficient enhanced k-means clustering algorithm. Journal of Zhejiang University-Science A 7(10):1626–1633
18. Motwani M, Arora N, Gupta A (2019) A study on initial centroids selection for partitional clustering algorithms. In Hoda MN, Chauhan N, Quadri SMK, Srivastava PR (eds), Software Engineering, pages 211–220, Singapore. Springer Singapore
19. Yedla M, Pathakota SR, Srinivasa TM (2010) Enhancing k-means clustering algorithm with improved initial center. International Journal of computer science and information technologies 1(2):121–125
20. Vadyala SR, Betgeri SN, Sherer EA, Amritphale A (2020) Prediction of the number of covid-19 confirmed cases based on k-means-lstm. arXiv preprint arXiv:2006.14752
21. Poompaavai A, Manimannan G (2019) Clustering study of indian states and union territories affected by coronavirus (covid-19) using k-means algorithm. International Journal of Data Mining And Emerging Technologies 9(2):43–51
22. Sonbhadra SK, Agarwal S, Nagabhushan P (2020) Target specific mining of covid-19 scholarly articles using one-class approach. arXiv preprint arXiv:2004.11706
23. Chinchorkar S (2020) Defining covid 19 containment zones using k-means dynamically
24. Aydin N, Yurdakul G (2020) Assessing countries' performances against covid-19 via wsidea and machine learning algorithms. Applied Soft Computing 97:106792
25. KUCUKEFE B (2020) Clustering macroeconomic impact of covid-19 in oecd countries and china. Ekonomi Politika ve Finans Araştırmaları Dergisi, 5(Özel Sayı):280–291
26. Zhang T, Lin G (2020) Generalized k-means in glms with applications to the outbreak of covid-19 in the united states. arXiv preprint arXiv:2008.03838
27. de la Fuente-Tomas L, Arranz B, Safont G, Sierra P, Sanchez-Autet M, Garcia-Blanco A, Garcia-Portilla MP (2019) Classification of patients with bipolar disorder using k-means clustering. PloS one 14(1):e0210314

28. Silitonga P (2017) Clustering of patient disease data by using k-means clustering. International Journal of Computer Science and Information Security (IJCSIS) 15(7):219–221
29. Das N, Iqbal MDA (2020) Nearest blood & plasma donor finding: A machine learning approach. In 2020 23rd International Conference on Computer and Information Technology (ICCIT), 1–6. IEEE
30. Alam MdS, Rahman MdM, Hossain MA, Islam MdK, Ahmed KM, Ahmed KT, Singh BC, Miah MdS (2019) Automatic human brain tumor detection in mri image using template-based k means and improved fuzzy c means clustering algorithm. Big Data and Cognitive Computing 3(2):27
31. Shi Y (2022) Advances in big data analytics: theory, algorithms and practices. Springer, Singapore
32. Arthur D, Vassilvitskii S (2006) k-means++: The advantages of careful seeding. Technical report, Stanford
33. Aloise D, Deshpande A, Hansen P, Popat P (2009) Np-hardness of euclidean sum-of-squares clustering. Machine learning 75(2):245–248
34. Berkhin P (2006) A Survey of Clustering Data Mining Techniques, 25–71. Springer Berlin Heidelberg, Berlin, Heidelberg
35. Abdi H, Williams LJ (2010) Principal component analysis. Wiley interdisciplinary reviews: computational statistics 2(4):433–459
36. Sehgal S, Singh H, Agarwal M, Bhasker V et al (2014) Data analysis using principal component analysis. In International Conference on Medical Imaging, m-Health and Emerging Communication Systems (MedCom), 45–48. IEEE
37. Altman DG, Bland JM (1994) Statistics notes: quartiles, quintiles, centiles, and other quantiles. Bmj 309(6960):996
38. Michigan State University Health Care. Mqic patient data 100k sample - visualizingvisualizing. https://www.visualizing.org/mqic-patient-data-100k-sample/, 2022. Accessed 1 May 2022
39. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12:2825–2830
40. Total covid-19 tests performed by country - humanitarian data exchange. https://data.humdata.org/dataset/total-covid-19-tests-performed-by-country, 2022. Accessed 1 May 2022
41. Roser M (2022) Covid-19 testing policies, sep 3, 2020. https://ourworldindata.org/grapher/covid-19-testing-policy?region=Asia. Accessed 1 May 2022
42. Roche Data Science Coalition. Uncover covid-19 challenge — kaggle. https://www.kaggle.com/roche-data-science-coalition/uncover, 2022. Accessed 1 May 2022
43. Coronavirus government response tracker — blavatnik school of government. https://www.bsg.ox.ac.uk/research/research-projects/coronavirus-government-response-tracker, 2022. Accessed 1 May 2022
44. Kodinariya TM, Makwana PR (2013) Review on determining number of cluster in k-means clustering. International Journal 1(6):90–95
45. Sarker IH (2022) Smart city data science: Towards data-driven smart cities with open research issues. Internet of Things, 100528
46. Tien JM (2017) Internet of things, real-time decision making, and artificial intelligence. Annals of Data Science 4(2):149–178