

# Comparison of Clustering Algorithms for Revenue and Cost Analysis

Nataliya Boyko, Solomiya Hetman and Iryna Kots

*Lviv Polytechnic National University, Profesorska Street 1, Lviv, 79013, Ukraine*

## Abstract

The modern period of development of society is characterized by a significant impact on its information technology in all spheres of life. Marketing is no exception. Among the great competition, it is so difficult to interest a potential buyer with your product. The solution to this problem is data clustering. The aim of the work is to investigate two popular clustering algorithms DBSCAN K-means, to analyze the dataset of customer data. The experiments confirmed the efficiency of the proposed methods for data clustering. It has been investigated that K-means for analyzing customer data. After all, the data are non-spherical in shape and have different densities, contain noise. However, the DBSCAN algorithm is excellent with such data.

## Keywords 1

Data Mining, Clustering, Clustering Algorithms, DBSCAN, K-means

## 1. Introduction

Clustering – is a unsupervised method of learning, that allows you to group a set of objects by different characteristics. The purpose of cluster analysis is to find existing structures, which is expressed in the formation of groups of similar objects – clusters. Clustering is needed to identify a structure in the data [1, 5].

In data mining problem with the help of cluster analysis we can create a comprehensive summary of data for classification, identify patterns, form and test a hypothesis. In addition, cluster analysis is often used to identify data that is "knocked out" among others, because such data correspond to points located at a distance from any cluster. Cluster analysis is also used to compress and summarize data [2, 7].

The purpose of clustering is to get new knowledge. In order to obtain reliable data and perform the analysis correctly, we need to choose the right clustering algorithm. Using clustering algorithms, we can find similarities between clients A and B and, based on this data, make recommendations to the client [3, 10, 14, 17]. Therefore, the main purpose of this article is to compare two known clustering algorithms. Investigate for which data it is better to use this or that algorithm. Compare the complexity and execution time of algorithms [8].

## 2. Review of literature sources

The main task for the analysis of data of different types is the problems of clustering. Their task is to identify features that work closely together to identify objects belonging to the same group (cluster). Clustering is unattended learning. The issue of clustering is considered in the article of scientists [1, 5, 8, 9]. In their article [17] for 2017, K. Chitra, Dr. D. Maheswari, they consider several clustering algorithms and generally analyze their work. In our paper, we provide a detailed overview

---

COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems, April 22–23, 2021, Kharkiv, Ukraine

EMAIL: nataliya.i.boyko@lpnu.ua (N. Boyko); solyahetman2013@gmail.com (S. Hetman); Iryna.i.kots@lpnu.ua (I.Kots)

ORCID: 0000-0002-6962-9363 (N. Boyko); 0000-0002-1096-465 (S. Hetman); 0000-0002-3008-436X (I.Kots)

© 2021 Copyright for this paper by its authors.

Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)



of two clustering algorithms (DBSCAN and K-tools) for data set analysis. Researchers wrote about them in their work [6, 10]. And also we investigate with what date this or that algorithm works better [12]. Researchers wrote about these optimization components in their works [12, 15, 17].

Consider in more detail the operation of the DBSCAN algorithm, which is based on determining the density between clusters. His job is to separate high and low density clusters [15, 17]. The authors [6,10,12, 17] describe in detail the algorithm K-means, gives a formal description, advantages and disadvantages.

In the works of researchers [8, 10] there are experiments that confirmed the effectiveness of the author's proposed methods of data clustering. In [11], it was investigated that K-tools for analyzing customer data. In addition, the researchers [8, 12], after analyzing the data, were not spherical, they wrote about it in their research. Thus, they wrote about different data densities and noise. Also in [17] was described the feasibility of using the DBSCAN algorithm, which also had the following data.

### 3. Proposed methodology

To perform the DBSCAN algorithm, you need to use two parameters that allow you to quickly group objects by characteristics into clusters [6, 9]:

1. Euclidean distance, which shows the distance between two points, it is denoted by EPS. This parameter determines the distance between adjacent points.

2. The second parameter is to form a dense cluster. It is determined by the minimum number of points.

Knowing these two parameters and how to determine them, you can divide the data points into certain categories:

The first category includes the main point, which takes the minimum number of points within the epsilon (MinPts) [4, 11, 17]. Next is the boundary point, which determines the path to the main point. The third category is the selected point, which is not associated with any dense clusters and is called the result.

Consider the algorithm of K-means. To implement it, you need to have the values of entry points and the value of K, where K is the number of required clusters. To analyze this method you need to follow a certain algorithm:

1. In the first step, you need to select the points K, which will act as the initial centroids.
2. The next step is to group relative to the centroid to create a cluster K. To do this, determine the Euclidean distance to each point relative to the centroid within the cluster.

If we have point  $p = (p_1, p_2)$  i  $q = (q_1, q_2)$ , then the Euclidean distance between the points  $d(p, q)$  will be determined by the formula 1.

$$dist(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \quad (1)$$

According to formula 2, the Euclidean distance is assigned to the points closest to the centroid. Where  $c_i$  centroids of every class, a  $x$  point from set (Formula 2).

$$\min dist(c_i, x) \quad (2)$$

3. Find next centroid, moving the centroid to the center of its cluster (Formula 3).

$$c_i = \frac{1}{|s_i|} \sum_{x_i \in s_i} x_i, \quad (3)$$

where  $s_i$  – the set of all points assigned to  $i$ -th cluster.

### 4. Experiments

First you need to examine the selected dataset «Mall\_Customer» (Table 1). The selected dataset contains data on customer age, gender, annual income, and average spending.

**Table 1**

Dataset «Mall\_Customer»

	Customer ID	Gender	Age	Annual Income(k\$)	Spending Score (1–100)
0	1	male	19	15	39
1	2	male	21	15	81
2	3	female	20	16	6
3	4	female	23	16	77
4	5	female	31	17	40

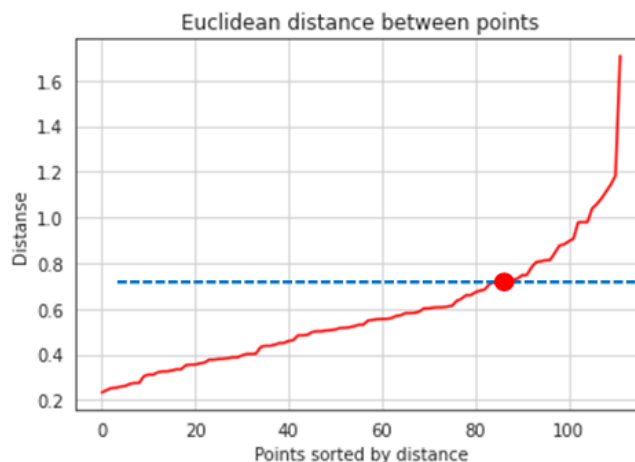
Using data, we tried to group the data and understand. Let's start the research with the DBSCAN algorithm.

As we know algorithm DBSCAN basically requires two parameters to search for clusters:

1. Eps;
2. MinPoints.

Our clustering depends on the values of these points.

The first step, to form a cluster requires a minimum number of points, which is denoted by MinPoints [13, 14]. It is generally accepted to choose MinPoints not less than 3. As the data set increases, the number of MinPoints increases. For our dataset we take MinPoints 4. In order to find eps, we need to calculate the Euclidean distance to n nearest neighbor of the point m. The next step is to plot the distances (Figure 1).



**Figure 1:** Euclidean distance between points

The Figure 1 shows the Euclidean distance between points, where the y – axis shows the distance between n – points, and the x – axis shows the points sorted by distance. The principle of eps selection is to select the point after which distance starts to increase. All points that will be located on the right side will be the noise. The number of noise will depend on the choice of eps. Therefore, according to Figure 2 eps = 0.7.

We found parameters that we need for this algorithm:

1. Eps – 0,7.
2. MinPoints – 4.

It should be noted that these parameters directly affect the result. The number of noise will depend on the choice of eps.

The DBSCAN algorithm begins its work by selecting one observation. Then counts the number of neighbors at a distance of eps (0.7). If the number of neighbors more than MinPoints (4), we classify our point as the basis and use it to expand the cluster. Let's look at the Figure 2, we can see that on a distance eps from the selected point are 6 points. So, we can say that the main point can form a cluster.



**Figure 2:** Basic algorithm. The first classified point

Each neighbor, which we have not considered before, we write in the cluster, and then use it as a point for further expansion of the cluster. Well, we begin to expand our cluster, consider the next neighbor and its neighbors (neighbors of the 2nd degree) and repeat the same algorithm. To expand the cluster, we consider the number of second – degree neighbors. If their number exceeds MinPoints, then the extension point becomes the main point that will determine the next extension. Looking at the figure we can see that at a distance eps from our point is 4. So, we can assume that our point belongs to our cluster (Figure 3).



**Figure 3:** Classified point

We continue our algorithm further. Choose the next point. We can see that there is one point left at the eps' distance, and we need at least 4 points to form a cluster, so our point should not be up to any class, and we classify as noise (Figure 4).



**Figure 4:** A point that we do not include in our cluster

And so we will continue until we visit will all the points. When we do not have enough neighbors and main points to expand, we have completed the classification of the cluster (Figure 5). And we get the first formed cluster.



**Figure 5:** The first cluster is formed

Then we select a new point and start the process again. We continue until we form all clusters. At the Figure 6 we can see how the DBSCAN algorithm divided our data into clusters.



**Figure 6:** Clustered data

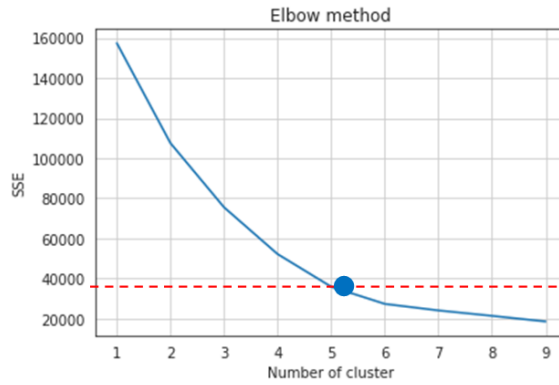
Now let's look at the operation of the K-means algorithm.

To implement the K-means algorithm you need to find the following values:

1. Number of cluster K.
2. K – centroids.

In algorithm K-means of big importance is the correct choice of the value of the parameter  $k$  – if the number of clusters is not known in advance.

In this case, it is usually considered  $k = \sqrt{n}$ , where  $n$  – sample size. However, for a large  $n$ , this choice of the parameter  $k$  will lead to a very large number of clusters, which will lead to incorrect clustering. Therefore, we will apply Elbow method. It implies multiple cyclic executions of the algorithm with an increase in the number of selected clusters, as well as subsequent deposition on the graph of the clustering score (Figure 7).



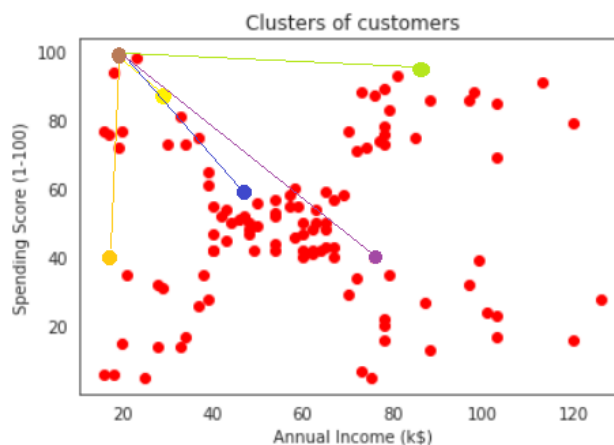
**Figure 7:** Elbow method

We need to choose the point after which the schedule stops falling sharply. In our case, it will be point 5. That is for our clustering we will use 5 clusters. The next step is to select K – centroid (Centroid is a point that is the center of the class). Well, randomly we choose 5 centers (Figure 8).



**Figure 8:** Initial selection of centers

We choose a point and look for the Euclidean distance from the selected point on the set of centroids (Figure 9).



**Figure 9:** Distances from a point to certain clusters

Next, we assign a given point to the cluster to which it is closest. The same we do for all points. When we have passed all the points, we look at the location of our centroids and move them to the

cluster of points. as a result, we obtain the location of new centroids. This process needs to be reused until we know a constant value for the centers, and the last cluster will be considered as the last solution of the cluster (Figure 10).



**Figure 10:** Clustered data

Our centers around which clusters are formed are highlighted in pink on the Figure 10.

## 5. Results

**Table 2**  
Comparison of results

Method	Preview
DBSCAN	
K-MEANS	

Comparing the work of algorithms, we can see that the given dataset is best handled by the algorithm K-means. We can that algorithm divided data on 5 clusters.

Table 2 shows five clusters. Let's analyze each of them: Individuals with low incomes and expenses are presented in cluster 4 and marked in purple. These individuals are not of interest for analysis because they are able to spend money within their income.

In the second cluster, which is depicted in orange, there are people who, despite low incomes, like to make reckless purchases.

In the cluster 0, the points of which are colored red, presents a sample of people with average incomes and expenses that correspond to this income.

In the cluster 1, the points of which are colored blue, shows the category of people who have high incomes and correspondingly high costs.

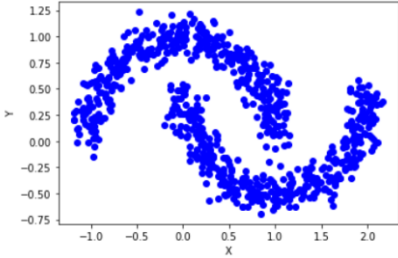
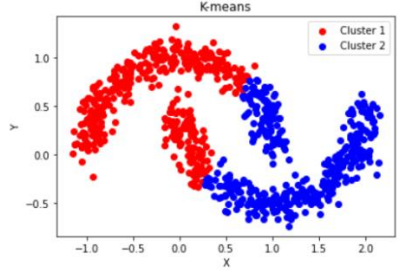
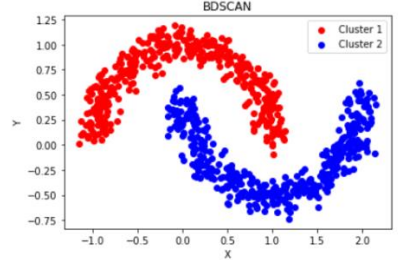
In the cluster 3, marked in yellow, represents a group of people who have high incomes but low costs. this is a category of buyers who are not satisfied with the quality of goods or services. Therefore, you need to focus on this category.

Therefore, after analyzing our data through clustering methods, we can conclude that to increase profits it is necessary to pay attention to clusters 3 and 5. People from clusters 1 and 2 are frugal, so they spend money wisely. Cluster 4 is people who do not have the opportunity to buy additional or expensive goods.

## 6. Discussions

Let's try to compare the work of our algorithms by applying to different data (Table 3).

**Table 3**  
Comparison of algorithms

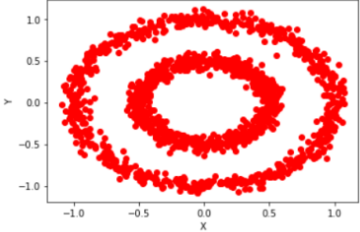
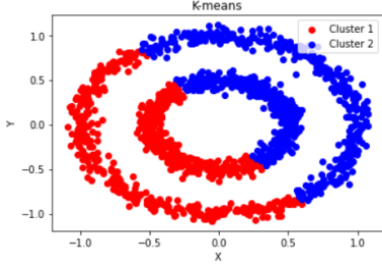
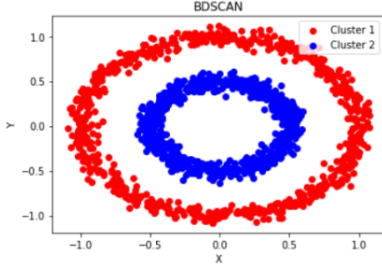
Method	Preview
Initial data set	
K-means	
DBSCAN	

We have a data set in the form of two crescents. We can see in the Table 3, that only the DBSCAN algorithm shown us the correct clusterization. Algorithms divided our data into 2 clusters, but only



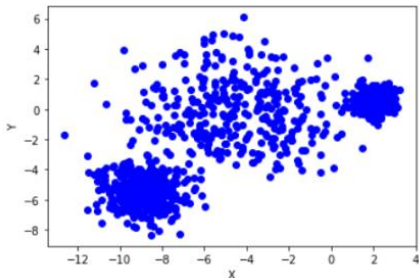
DBSCAN divided the data correctly. The K-means algorithm divided our data set according to the location of the centroid. And as we know that our centroids are shifted toward the largest cluster, we can conclude that the K-means algorithm works poorly when the data is not spherical (Table 4).

**Table 4**  
Comparison of algorithms

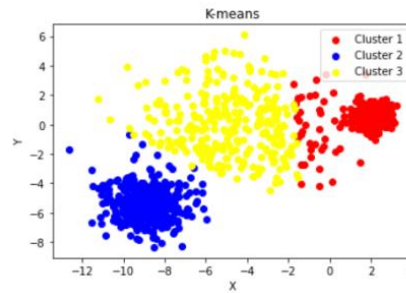
Method	Preview
Initial data set	
K-means	
DBSCAN	

We have the following data set. The DBSCAN and K – means algorithms identified 2 clusters (Table 4). However, again, only DBSCAN coped with the task correctly. First of all, K-means works badly with non-spherical data. And also the algorithm divides the data into approximately identical clusters (Table 5).

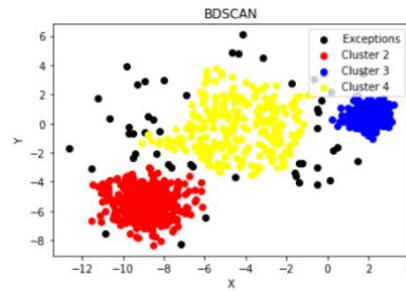
**Table 5**  
Comparison of algorithms

Method	Preview
Initial data set	

K-means



DBSCAN



Well, let's consider the following example. In general, we can say that each of the algorithms has done the task correctly (Table 5). However, the DBSCAN algorithm coped with the task better. To form a dense cluster, you need to determine the minimum number of points. For this purpose the point was entered in a certain group. If the point does not belong to any cluster, it is defined as noise. Accordingly, our algorithm divided our data into 3 clusters on the allocated emissions.

Let's compare the dependence of the running time of algorithms on the amount of data (Table 6).

**Table 6**  
Dependence of K-means operating time on the amount of data

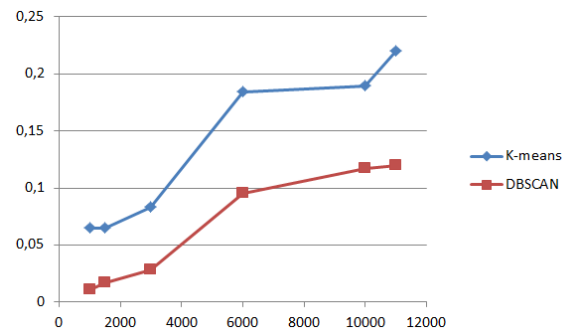
The amount of observation, units	Work time, seconds
1000	0.065
1500	0.065
3000	0.083
6000	0.184
10 000	0.189
11 000	0.22

We can see in the Table 6 that a large amount of data for K-means is not a problem, because the data processing time is not high. After all, even when we increased the amount of data 11 times, the operating time of the algorithm did not increase very sharply, the time difference is only 0.15 seconds.

**Table 7**  
Dependence of DBSCAN operating time on the amount of data

The amount of observation, units	Work time, seconds
1000	0.011
1500	0.017
3000	0.028
6000	0.095
10 000	0.117
11 000	0.12

The DBSCAN algorithm also quickly classifies data (Table 7). It handles a data set that contains 11,000 observations in just 0.12 seconds.



**Figure 11:** Dependence of time on the amount of data

We can see that the fastest is the DBSCAN algorithm, in second place is the K-means algorithm (Figure 11).

## 7. Conclusions

K-means clustering for analysis of customer data in order to allocate clusters.

Advantage:

- Easy to implement.
- Suitable for a large data set and is calculated much faster than on smaller data sets.
- Is fast and efficient in terms of computational cost.
- This clustering algorithm works well when working with spherical clusters. It works with the assumption of a common distribution of features, because each cluster is spherical.

Disadvantages:

- To implement the algorithm you need to know the number of clusters. Accordingly, we need for additional analysis, for example, to use the Elbow method to find the number of clusters. Accordingly, if we do not correctly determine the number of clusters, our algorithm will work incorrectly.
- In data should not be any noisy (data that differ from the total sample of data).
- Each cluster should have approximately the same number of observations.
- Data must have a spherical distribution, the same data density.
- There is a problem that is the "curse of dimension" at large dimensions, because we use a measure of distance.
- Fast.

DBSCAN clustering for analysis of customer data in order to allocate clusters.

Advantage:

- The algorithm handles data that contains noise. Due to the fact that we specify the minimum number to form a dense cluster. And we divide our points into three groups.
- The algorithm can be executed without determining the number of clusters.
- This algorithm allows you to work with a set of data of different forms. The MinPts parameter determines the effect of a single bond, which allows you to find a cluster that is not related to another.
- Fast.

Disadvantages:

- It is very sensitive to changes in parameters (MinPoints and EPS) and, accordingly, a small change in parameters can lead to a big change in results.
- Cluster datasets with large density differences cannot be well established because the combination of MinPoints and EPS cannot be properly selected for all clusters.

## 8. References

- [1] A.K. Tung, J. Hou, J. Han, Spatial clustering in the presence of obstacles, in: The 17th Intern. conf. on data engineering (ICDE'01), Heidelberg, 2001, pp. 359–367.
- [2] C. Boehm, K. Kailing, H. Kriegel, P. Kroeger, Density connected clustering with local subspace preferences, in: Proc. of the 4th IEEE Intern. conf. on data mining, IEEE Computer Society, Los Alamitos, 2004, pp. 27–34.
- [3] N. Boyko, K. Boksho, Application of the Naive Bayesian Classifier in Work on Sentimental Analysis of Medical Data, in: The 3rd International Conference on Informatics & Data – Driven Medicine (IDDM 2020), Växjö, Sweden, November 19 – 21, 2020, pp. 230 – 239.
- [4] D. Guo, D.J. Pequet, M. Gahegan, ICEAGE: Interactive clustering and exploration of large and high-dimensional geodata, vol. 3, N. 7, Geoinformatica, 2003, pp. 229 – 253.
- [5] D. Harel, Y. Koren, Clustering spatial data using random walks, in: Proc. of the 7th ACM SIGKDD Intern. conf. on knowledge discovery and data mining, San Francisco, California, 2000, pp. 281–286.
- [6] D.J. Pequet, “Representations of space and time”. N. Y.: Guilford Press (2000)
- [7] H.-Y. Kang, B.-J. Lim, K.-J. Li, P2P Spatial query processing by Delaunay triangulation, Lecture notes in computer science, vol. 3428, Springer/Heidelberg, 2005, pp. 136–150.
- [8] M. Ankerst, M. Ester, Kriegel H.-P., Towards an effective cooperation of the user and the computer for classification, in: Proc. of the 6th ACM SIGKDD Intern. conf. on knowledge discovery and data mining, Boston, Massachusetts, USA, 2000, pp. 179–188.
- [9] C. Zhang, Y. Murayama, Testing local spatial autocorrelation using, vol. 14, Intern. J. of Geogr. Inform. Science, 2000, pp. 681–692.
- [10] N. Boyko, B. Mandych, Technologies of Object Recognition in Space for Visually Impaired People, in: The 3rd International Conference on Informatics & Data-Driven Medicine (IDDM 2020), Växjö, Sweden, November 19 –21, 2020, pp. 338 – 347.
- [11] R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan, Automatic sub-space clustering of high dimensional data, vol. 11(1), Data mining knowledge discovery, 2005, pp. 5–33.
- [12] V. Estivill-Castro, I. Lee, Amoeba: Hierarchical clustering based on spatial proximity using Delaunay diagram, in: 9th Intern. Symp. on spatial data handling, Beijing, China, 2000, pp. 26–41.
- [13] I. Turton, S. Openshaw, C. Brunson, Testing spacetime and more complex hyperspace geographical analysis tools, Innovations in GIS 7, London: Taylor & Francis, 2000, pp. 87–100.
- [14] N. Boyko, N. Tkachuk, Processing of Medical Different Types of Data Using Hadoop and Java MapReduce, in: The 3rd International Conference on Informatics & Data-Driven Medicine (IDDM 2020), Växjö, Sweden, November 19 – 21, 2020, pp. 405 – 414.
- [15] C. Aggarwal, P. Yu, Finding generalized projected clusters in high dimensional spaces, in: Intern. conf. on management of data, ACM SIGMOD, 2000, pp. 70–81.
- [16] C.M. Procopiuc, M. Jones, P.K. Agarwal, T.M. Murali, A Monte Carlo algorithm for fast projective clustering, in: Intern. conf. on management of data, ACM SIGMOD, Madison, Wisconsin, USA, 2002, pp. 418–427.
- [17] K. Chitra , Dr. D.Maheswari, A Comparative Study of Various Clustering Algorithms in Data Mining. International Journal of Computer Science and Mobile Computing, Vol.6 Issue.8, August 2017, pp. 109 –115.