DOI 10.29042/2018-3399-3405

Helix Vol. 8(3):3399-3405

An Efficient Clustering Approach using DBSCAN

[1] R. V. Sravana Kumar, [2] Dr. S. Sreenivasa Rao, [3] Dr. P. Srinivasrao
M.V.G.R College of Engineering, Vizianagaram
Email: *rvsravankumar04a6@gmail.com*

Received: 3rd March 2018, Accepted: 19th March 2018, Published: 30th April 2018

Abstract
With the fast development of the Internet, the information available in the database showing explosive growth where it wouldn't be that easy to obtain desired information accurately from the available data. This unexpected growth of internet has also led to the frequent usage of e-commerce sites which concentrate on sales based on their relevant products. To do so clustering of products is done so as to identify the sales with profits and which are lagging behind by finding similarities among the products. The clustering analysis is done with the help of various clustering algorithms like DBSCAN, K-means, GRID based, Hierarchical based, etc., .Each algorithm will produce different results; you'll never be certain whether one result is better than the other or even whether the result is of any value. K-means algorithm is preferable choice for datasets having small number of clusters with proportional sizes and linearly separable data and also you can scale it to use on very large data sets. K-means algorithm needs the number of clusters fixed in advance and it arises problems when the data contains outliers. Also the downside of K-means is it leads to problems when clusters are of different sizes, densities and non-globular shapes. The DBSCAN implementation does not require any user-defined initialization parameters to create an instance like K-means do. So DBSCAN algorithm is more advantageous when compared to other algorithms. It is because DBSCAN doesn't need one to specify the amount of clusters within the knowledge a priori, as against K-means. DBSCAN will notice different arbitrary formed clusters. Density clustering (DBSCAN) seems to correspond more to human intuitions of clustering, rather than distance from a central clustering point (K-means).

Introduction

Massive growth of data available in database is taking place with increase in development of internet and its usage[1]. There are many applications with massive amount of data which are creating limitation in data storage capacity and processing time. Considering such huge amount of data it is not possible to obtain desired information accurately from this available data. In such scenario of growing data, business models like e-commerce has been evolving rapidly. This rapid evolution of e-commerce has few requirements like: analyzing the sales, demand-manufacture ratio, establishing new business models based on market fluctuations/scenario for given goods.

For example, consider in an e-commerce site we want to improve our sales by recommending relevant products to our customers[2]. In such case initially the company must need to know about the sale of their products so as to check which product is hiking in sales with good benefits and profits and which are lagging behind[3]. But this type of process wouldn't show better results all the time. Find clusters based on the products that the users have bought and by this clusters we find similarities between customers .

E-commerce business and retailers work in a competitive and fast paced environment with the online presence these days even though influenced by cost and online advertisements. For separate information sources and data needs conventional approaches were divided with individual analytical tools. Gathering the information along is often finished manually, a human operated method. Decisions are consequentially suboptimal owing to the volume, variety and velocity of information. The challenge here is not only about gathering data but also analyzing and using it appropriately. Also it can be difficult for linking, matching, correlating and interpreting the data obtained from different sources. These are the main emphasis of clustering analysis where it assures results acquired by superior data.

Chetan Dharni and Meenakshi Bnasal has proposed an algorithm DBSCAN is developed to identify the spatial data clusters with noise. In this paper, steps are used to evaluate the clusters which are Read Input Dataset, Select tool, Apply Algorithm, Calculate Parameters, Result fetching, Plot graphs. The algorithm which is proposed is used to stream these all datasets by using WEKA tool. The proposed method efficiently examine the clusters for huge datasets. The Advantage of these algorithm is more scalable. **Asieh Ghanbarpour and Behrooz Minaei** has proposed ExDBSCAN algorithm instead of normal DBSCAN algorithm with two inputs those are, EPS and Minpts . The ExDBSCAN takes only one input i.e., Minpts but Ɛ value was assumed as a small value and increase the radius of the cluster progressively to obtain exact neighborhood. In this aspect ,the outlier may be marked as a constituent of cluster. To eliminate this problem ,the paper was used statistical summary of data points for detecting of outliers. The advantage of the ExDBSCAN is, it can detect multi density cluster in a dataset. The proposed method is accomplished in the java programming by

using Eclipse environment. This method cover the drawbacks of DBSCAN and OPTICS. **Mohammad F. Hassanin et.al** has developed DDBSCAN to overcome the drawbacks in DBSCAN algorithm . The DBSCAN algorithm fails to obtain different density clusters, adjacent clusters and few noise points .Then provide a density threshold, it tells the given point is joining in the cluster or not. DDBSCAN acquirs two parameters which are Eps and threshold. Eps is determine the all objects within the region and threshold is used for decision criteria. **Jianbing Shen et.al** has proposed a real time image super pixel segmentation method with 50fps by making use of DBSCAN algorithm. Two step framework was used in this paper for reducing the computational cost of the super pixel algorithm. The two steps are ,first one is clustering stage and other step is merging stage. In the clustering stage, the DBSCAN algorithm to form a cluster by using similarity of color. In the merging stage, clusters which are small were merged into super pixel by their neighborhood. The performance increased by using super pixel segmentation with DBSCAN algorithm. **Irving Cordova and Teng-Sheng Moh** has proposed a new algorithm based on DBSCAN using the Resilient Distributed Datasets approach, that is RDD DBSCAN. This RDD DBSCAN is more scalable than normal DBSCAN, it is overcome the scalability problems by operating in distributed manner. The limitation of these algorithm was cost of the communication among the computational nodes. **Madhuri Debnath et.al** has proposed a novel density based spatial clustering algorithm called as K-DBSCAN. This algorithm mainly concentrates for identifying the cluster points with similar spatial density. The advantage of K-DBSCAN is finding arbitrary shaped clusters in variable density regions. The K-DBSCAN algorithm works under two phases ,which are initially data objects are split into different density levels and it extracts the clusters using a modified version of the DBSCAN. **LI Meng'ao et.al** has proposed a modified DBSCAN algorithm based on grid cells, it is overcome the drawback of the normal DBSCAN algorithm ,that is time cost of the algorithm is more. These algorithm modifies the most time consuming region query process of DBSCAN and reduces innumerable unnecessary query operations by dividing data space into grid cells. The advantage of these proposed algorithm has a higher accuracy and lower time complexity. **Thanh N. Tran et.al** has proposed a modified version of DBSCAN algorithm to solve the complication in normal DBSCAN algorithm, that is the normal DBSCAN algorithm will become unstable while detecting border objects of adjacent clusters. The proposed algorithm is better performance than normal DBSCAN algorithm. **Satyasai Jagannath Nanda and Ganapati Panda** has planned a different strategy to scale back the computational complexity related to DBSCAN by expeditiously implementing new merging criteria at the initial stage of evolution of clusters. The new density based clustering algorithm planned considering coefficient of correlation as a similarity measure . **Damodar Reddy Edla and Prasanta K. Jana** has planned a completely unique based mostly DBSCAN technique to cluster the gene expression data for covering the demerits of DBSCAN. The most downside of the DBSCAN algorithmic program is it's computational complexity to resolve this downside the prototypes created from a square error bunch technique like K-means. These algorithmic has higher interpretation than existing algorithms. **Abir Smiti and Zied Elouedi** has proposed a density based approach DBSCAN-GM seen advantages over explicit samples , it discover the number of clusters and detects the noises. In this paper ,improve the already define crisp DBSCAN-GM algorithm into soft DBSCAN-GM. It deals with soft objects and it is combined with the fuzzy set theory. The advantage of these algorithm gives the effectiveness and perform good quality of clustering. **P. Viswanath and V. Suresh Babu** has proposed a hybrid clustering technique called rough-DBSCAN and it is analyzed using rough set theory. In general DBSCAN has a higher time complexity, so it is not a better approach for large datasets. For this reason , a solution is presented in these paper so as to apply the leader clustering method and obtain the prototype called leader from the dataset . These algorithm is faster than the normal DBSCAN. **Fabio Baselice et.al** has proposed a novel segmentation approach based on two innovations. For these innovations makes the algorithm particularly robust and allows the classification of identical segments. The technique is used based on Euclidean distance ,it is improve the correct classification rate. **Atrayee Dhua et.al** has presents comparison between two density based clustering algorithms, which are DBSCAN and Mean Shift .In this paper, image as a dataset of pixels and this paper first removes noise by using median filtering technique, then apply DBSCAN algorithm into it and find the clusters in the image. **Karlina Khiyarin Nisa et.al** has developed a web application ,these application conducts clustering on the hotspot data. These application implemented in the DBSCAN algorithm with the use of shiny web framework for R programming. In this paper, these hotspots are related to forest regions of south sumathra and kalimantan.

Materials and Methods

Density-based spatial clustering of applications with noise (DBSCAN) [4] may be a data clustering algorithm which is a density-based. Consider a group of points in some area, it teams along points that area unit closely packed along, marking as outliers points that lie alone in low-density regions. DBSCAN needs 2 parameters: ε (eps) and therefore the minimum variety of points needed for making a dense region (minPts). It begins with degree or starting point that was not visited. This point's ε-neighborhood is obtained, and if there exist sufficiently several points, a cluster is started. If not it is labeled as noise. At certain extent if there exist dense part of cluster then its ε-neighborhood is considered additionally a part of that cluster. Therefore all points which are found at intervals the ε-neighborhood are included as their own ε-neighborhood once they are dense. This method extends till the density-connected cluster is totally found. Then, a replacement unvisited purpose is obtained and processed, resulting in the invention of an extra cluster or noise.

DBSCAN algorithm is more advantageous when compared to other algorithms. It is because DBSCAN doesn't need one to specify the amount of clusters within the knowledge a priori, as against k-means. DBSCAN will notice different arbitrary formed clusters. It will also notice a cluster fully enclosed by unique cluster. Attributable to the MinPts parameter, the alleged single-link impact is minimized. DBSCAN incorporates a notion of noise, and is strong to outliers. DBSCAN needs simply 2 parameters and is basically unaffected to the ordering of the points within the info. DBSCAN is meant to be used with databases which will accelerate region queries, e.g. using of R* tree. The parameters minPts and ε is set by a site professional, if the information is well perceived. (DBSCAN) looks to correspond a lot of to human intuitions of cluster, instead of distance from a central clustered (K-Means).

DBSCAN[20] plays efficient role in all these compared to K-means algorithm because DBSCAN does not need one to outline the number of clusters within the knowledge a priori, as against k-means. DBSCAN will notice arbitrarily formed clusters. It will even notice a cluster fully enclosed by a unique cluster.

DBSCAN can be used with any distance function. The distance function (dist) can therefore be seen as an additional parameter.

The algorithm can be explained with pseudo code as follows[19]:

```
DBSCAN(D, epsilon, min_points):
C = 0
for each unvisited point Pin dataset
mark P as visited
sphere_points = regionQuery(P, epsilon)
if sizeof(sphere_points) < min_points
ignore P
else
C = next cluster
expandCluster(P, sphere_points, C, epsilon,
min_points)
```

$$\text{purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_{k} \max_{j} |\omega_k \cap c_j|$$

```
expandCluster(P, sphere_points, C, epsilon,
min_points):
add P to cluster C
for each point P in sphere_points
if P' is not visited
mark P' as visited
sphere_points' = regionQuery(P', epsilon)
if sizeof(sphere_points') >= min_points
sphere_points = sphere_points joined with
sphere_points'
if P' is not yet member of any cluster
add P' to duster C
regionQuery(P, epsilon):
return all points within the n -dimensional sphere
centered at P with radius epsilon (including P)
```

4. Environment Setup:

Our experiments were performed using two individual datasets and java 1.6 is installed where using DBSCAN algorithm the data sets are run using this java platform.

5. Results: We can apply DBSCAN algorithm with the Benchmark Dataset ,the following results will be observed.

Give t minimum points =5 and Threshold Distance =1000 such that the total clusters obtained is 25 for the data set 1 and for the dataset 2 , the given values are minimum points=1 and Threshold Distance =5.

Based on these values "Cluster purity measurements" is done. The measurements included are Purity, NMI[22], Rand Index[22], F1 measure[22], Homogeneity[21], Completeness[21]

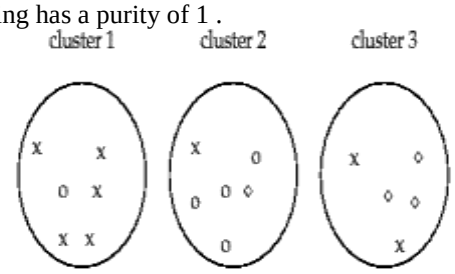
V-measure[21]. **Purity** [18] is a simple and transparent evaluation measure. In order to evaluate *purity* , each cluster is assigned to the class which occurs more frequently in the cluster After counting the number of correctly assigned documents and dividing by N accuracy of the assignment is finally measured.
Formally:

$$\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$$

where is the set of

$$\mathbb{C} = \{c_1, c_2, \ldots, c_J\}$$

clusters and is the set of classes. We interpret w_k as the set of documents in w_k and c_j as the set of documents in c_j in the given equation.
Figure below describes how to evaluate purity. Bad clusterings h+ave purity values close to 0, a perfect clustering has a purity of 1 .



▶ Figure 16.1 Purity as an external evaluation criterion for cluster quality. Majority class and number of members of the majority class for the three clusters are: x, 5 (cluster 1); o, 4 (cluster 2); and ◇, 3 (cluster 3). Purity is $(1/17) \times (5+4+3) \approx 0.71$.

When the number of clusters is, it is easy to achieve the High purity - in particular, purity is 1 if each of the document has its own cluster. Hence, purity couldn't be used to commutate the standard of the clustering opposing the number of clusters.

NMI(Normalized Mutual Information):
NMI is a good measure for determining the quality of clustering. It is an external measure because we need the class labels of the instances to determine the NMI. Since it's normalized we are able to calculate and compare the NMI between completely different clusterings having different range of clusters.

- Normalized Mutual Information:

$$NMI(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]}$$

where,
1) Y = class labels
2) C = cluster labels
3) H(.) = Entropy
4) I(Y;C) = Mutual Information b/w Y and C

Note: All logs are base-2.

$$H(.) = -\sum_{i=1}^{w} p\log(p)$$

Rand Index:
The **Rand index** or **Rand measure** in data clustering is a measure of the similarity between two data clusters. From a mathematical stand, Rand index is expounded to the accuracy, however is applicable even once category labels don't seem to be used.

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

Dataset	TP	FP	FN	TN
Employee	5	0	6	5
US Census data	2	0	1	1

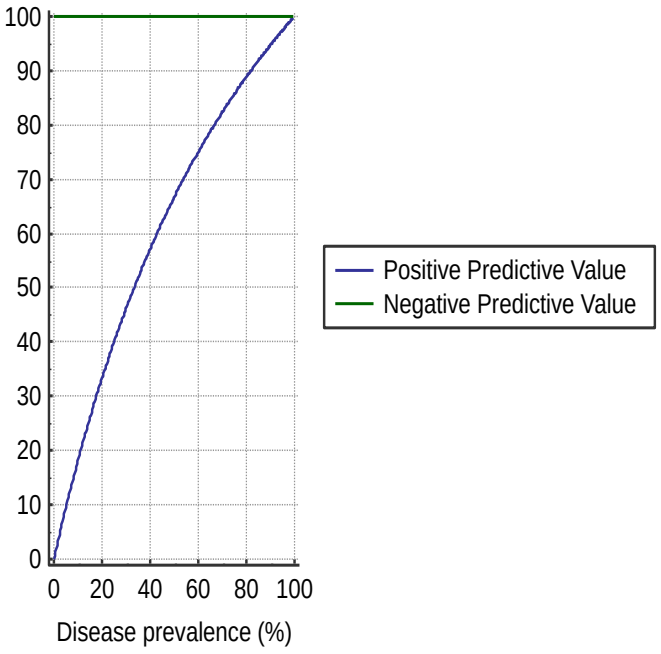


Figure 1

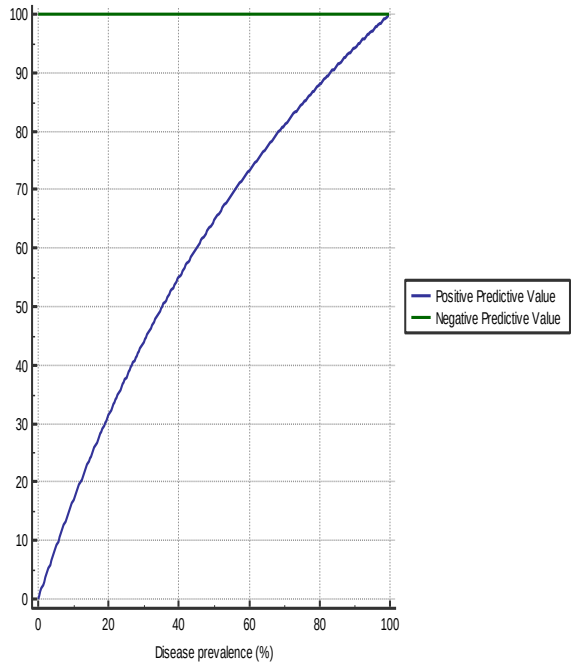


Figure 2

F measure(F_1):
The F_1 score is the balanced average of the "precision and recall", where an F_1 score reaches its best value at 1 and worst at 0.

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Precision= $\frac{TP}{TP+FP}$ Recall= $\frac{TP}{TP+FN}$

Homogeneity: For satisfying homogeneity criteria clustering has to assign **only** those data points that belong to single class in a single cluster.

Completeness: For satisfying completeness clustering should assign **all** those data points that belong to single class in a single cluster.

V-measure:
V-measure is used to explicitly measure to show how satisfactorily the aspect of homogeneity and completeness have been done. Where "V" indicates **validity**.

V=(2*Homogeneity*Completeness)(Homogeneity+ Completness)

Clustering result of the Employee Dataset is

Cluster	**Size**	**Distribution**	**Purity**
C1	22	21	0.954
C2	17	17	1
C3	23	14	0.608
C4	10	10	1
C5	5	5	1
C6	5	5	1
C7	8	8	1
C8	9	9	1
C9	7	7	1
C10	9	9	1
C11	7	6	0.8571
C12	7	7	1
C13	16	16	1
C14	9	9	1
C15	6	4	0.666
C16	8	8	1
C17	13	13	1
C18	6	5	0.833
C19	5	5	1
C20	6	6	1
C21	6	6	1
C22	11	5	0.4545
C23	5	5	1
C24	5	5	1
C25	5	5	1

Table 1: Calculation of Purity Measure

H(Y)	H(C)	H(Y|C)	I(Y,C)	NMI(Y,C)
1.3	1.34	0.06	1.24	0.94

Table 2: Calculation of NMI

Precision	Recall	F-Measure
1	0.454	0.625

Table 3: Calculation of F-Measure
Clustering result of the US Census Dataset is

Clusters	Size	Distribution	Purity
C1	95	92	0.968
C2	3	2	0.666
C3	2	2	1
C4	1	1	1
C5	3	2	0.666
C6	1	1	1
C7	5	5	1
C8	1	1	1
C9	3	3	1
C10	1	1	1
C11	1	1	1
C12	1	1	1
C13	1	1	1
C14	1	1	1

Table 4: Calculation of Purity Measure

H(Y)	H(C)	H(Y|C)	I(Y,C)	NMI(Y,C)
0.72	0.426	0.23	0.49	0.8551

Table 5: Calculation of NMI

Precision	Recall	F-Measure
0.666	0.666	0.666

Table 6: Calculation of F-Measure
Finally the results are obtained by computing these measures. It is shown in following table

Dataset	Purity	NMI	Rand Index	F measure	Homogeneity	Completeness (Normalized values(0-1))	V -Measure
Employee	0.934	0.94	0.625	0.625	0.06	0.166667	0.12
US Census data	0.9501	0.8551	0.50	0.666	0.23	0.43478	0.4388

Table 7: Comparison of The Datasets The tabular values are represented in graphical manner shown in below figure

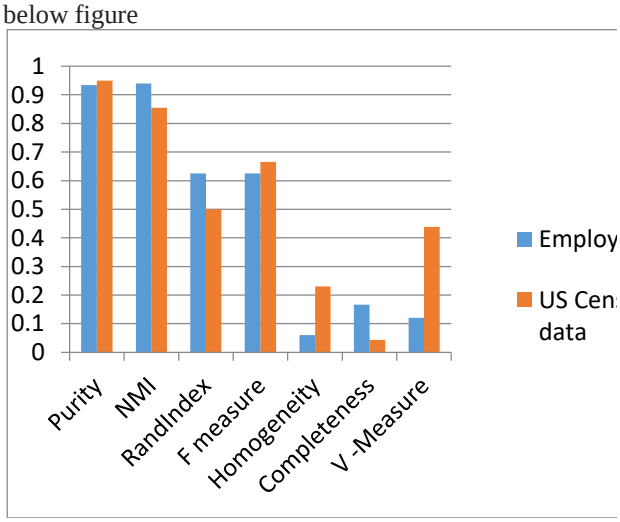


Figure 3

Conclusion

This article experimented on Employee and US census data Datasets using Data mining clustering technique called DBSCAN, and the Results are Evaluated using Cluster metrics such as Purity, NMI, Rand Index, F-measure, Homogeneity, Completeness, V-measure. This methods improves the Grouping quality.

References

1. Hassanin, Mohammad F., Mohamed Hassan, and Abdalla Shoeb. "DDBSCAN: Different Densities-Based Spatial Clustering of Applications with Noise." Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2015 International Conference on. IEEE, 2015.
2. https://towardsdatascience.com/how-dbscan-works-and-why-should-i-use-it-443b4a191c80
3. Bijuraj, L. V. "Clustering and its Applications." *Proceedings of National Conference on New Horizons in IT-NCNHIT*. 2013.
4. Dharni, Chetan, and Meenakshi Bnasal. "An improvement of DBSCAN Algorithm to analyze cluster for large datasets." Innovation and Technology in Education (MITE), 2013 IEEE International Conference in MOOC. IEEE, 2013.
5. Ghanbarpour, Asieh, and Behrooz Minaei. "EXDBSCAN: An extension of DBSCAN to detect clusters in multi-density datasets." Intelligent Systems (ICIS), 2014 Iranian Conference on. IEEE, 2014.
6. Shen, Jianbing, et al. "Real-Time Superpixel Segmentation by DBSCAN Clustering Algorithm." IEEE Transactions on Image Processing 25.12 (2016): 5933-5942.
7. Cordova, Irving, and Teng-Sheng Moh. "Dbscan on resilient distributed datasets." High Performance Computing & Simulation (HPCS), 2015 International Conference on. IEEE, 2015.
8. Debnath, Madhuri, Praveen Kumar Tripathi, and Ramez Elmasri. "K-DBSCAN: Identifying Spatial Clusters with Differing Density Levels." Data Mining with Industrial Applications (DMIA), 2015 International Workshop on. IEEE, 2015.
9. Meng'Ao, Li, et al. "Research and Improvement of DBSCAN Cluster Algorithm." Information Technology in Medicine and Education (ITME), 2015 7th International Conference on. IEEE, 2015.
10. Tran, Thanh N., Klaudia Drab, and Michal Daszykowski. "Revised DBSCAN algorithm to cluster data with dense adjacent clusters." Chemometrics and Intelligent Laboratory Systems 120 (2013): 92-96.
11. Nanda, Satyasai Jagannath, and Ganapati Panda. "Design of computationally efficient density-based clustering algorithms." Data & Knowledge Engineering 95 (2015): 23-38.
12. Edla, Damodar Reddy, Prasanta K. Jana, and IEEE Senior Member. "A prototype-based modified DBSCAN for gene clustering." Procedia Technology 6 (2012): 485-492.
13. Smiti, Abir, and Zied Elouedi. "Fuzzy density based clustering method: Soft

DBSCAN-GM." Intelligent Systems (IS), 2016 IEEE 8th International Conference on. IEEE, 2016.
14. Viswanath, P., and V. Suresh Babu. "Rough-DBSCAN: A fast hybrid density based clustering method for large data sets." Pattern Recognition Letters 30.16 (2009): 1477-1488.
15. Baselice, Fabio, et al. "A DBSCAN based approach for jointly segment and classify brain MR images." Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE. IEEE, 2015.
16. Nisa, Karlina Khiyarin, Hari Agung Andrianto, and Rahmah Mardhiyyah. "Hotspot clustering using DBSCAN algorithm and shiny web framework." Advanced Computer Science and Information Systems (ICACSIS), 2014 International Conference on. IEEE, 2014.
17. https://archive.ics.uci.edu/ml/machine-learning-databases/census1990-mld
18. https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html
19. https://en.wikipedia.org/wiki/DBSCAN.
20. https://blog.dominodatalab.com/topology-and-density-based-clustering
21. Rosenberg, Andrew, and Julia Hirschberg. "V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure." *EMNLP-CoNLL*. Vol. 7. 2007.
22. Lamari, Yasmine, and Said Chah Slaoui. "Clustering categorical data based on the relational analysis approach and MapReduce." *Journal of Big Data* 4.1 (2017): 28.

