Efficient Genetic K-Means Clustering for Health Care Knowledge Discovery

By: Rachelli Adler and Esther Malka Nusbacher
adviser: guy kelman
Jerusalem college of technology
Git: https://github.com/RachelliAA/K-clustering
Date: 07/08/2025

Cite of the article:
A. Alsayat and H. El-Sayed, "Efficient genetic K-Means clustering for health care knowledge discovery," 2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA), Towson, MD, USA, 2016, pp. 45-52, doi: 10.1109/SERA.2016.7516127.

keywords: {Clustering algorithms;Data mining;Algorithm design and analysis;Genetics;Diseases;Support vector machines;Self Organizing Map;K-Means;Genetic Algorithm;Clustering;Health care;Data Mining},

Table of contents:

Summary of the article:....2
1. Database preparation:....2
1.1 Liver dataset:.... 3
1.2 Heart dataset:.... 4
2. Baseline Clustering:....6
2.1 Kmeans....6
2.2 DBSCAN....7
3. Advanced clustering:.... 8
3.1 SOM....8
som_grid <- somgrid(xdim = 6, ydim = 6, topo = "hexagonal")....9
set.seed(123)....9
som_model <- som(X_scaled, grid = som_grid, rlen = 750, alpha = c(0.5, 0.01))....9
# X_scaled- Normalize features (scale to mean=0 sd=1)....9
3.2 genetic algorithm.... 9
3.2 K-means.... 9
4. Evaluate performances:....10
4.1 Figure 2: Heatmap....10

4.3 Figure 4: Density plot... 11
4.4 results:...11
Conclusion:.. 11
Bibliography:...12

Link to the article:
https://ieeexplore.ieee.org/document/7516127

Liver dataset:
https://archive.ics.uci.edu/dataset/60/liver+disorders

Heart dataset:
Cleveland Clinic Foundation Heart Disease

Summary of the article:

Data mining and machine learning are important tools in healthcare because they help us make better decisions, reduce the cost of patient care, identify groups of patients with similar conditions, detect causes of diseases, assist in clinical decision-making and healthcare policy development.

Clustering is useful for grouping patients based on things they have in common, which can help with choosing the right treatments. Patients with similar symptoms would have similar treatments.

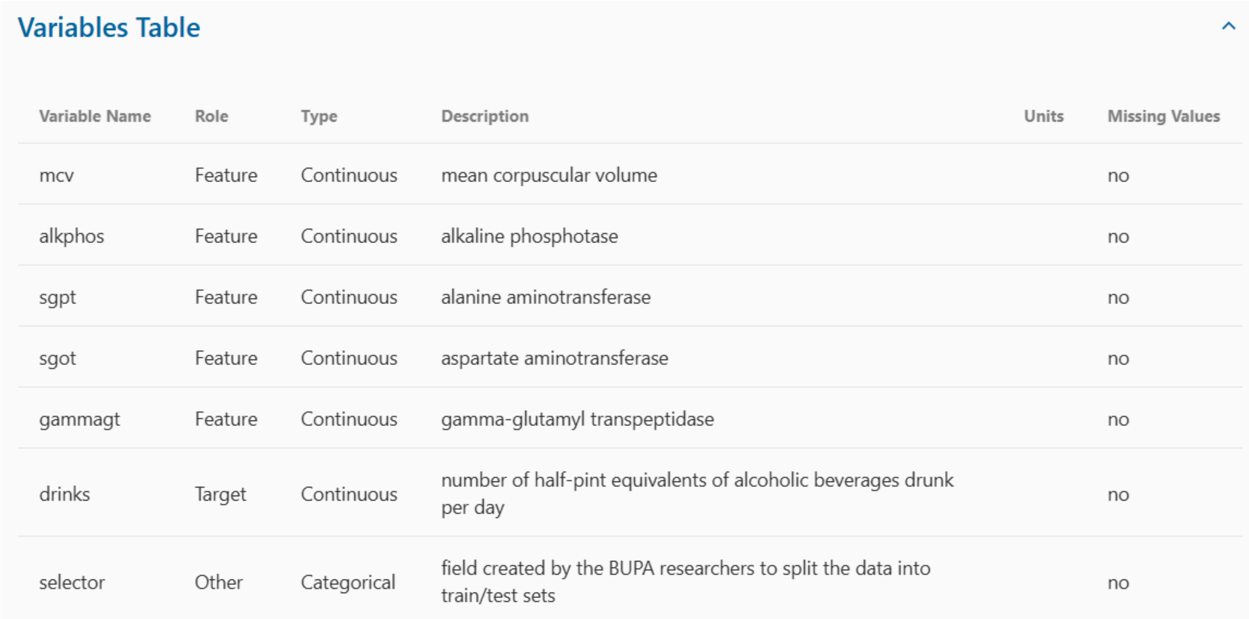
But the problem is, regular clustering methods like K-Means don't always pick the best number of groups, so the results aren't always accurate. Poor initialization of cluster centroids can lead to suboptimal results.
The article suggest "We propose an **efficient K-Means** clustering algorithm which uses the **SOM** method to discover the optimal segments number in the data as a preprocessing step"

1. Database preparation:

We used two datasets: heart disease from UCI Machine Learning Repository. Liver disease dataset from BUPA Medical Research Ltd
We googled them and and with luck we found them. We downloaded them.
*we were not given any code, or hints to where to find code. Hence ALL the code was written by us with the help of chatGPT.

1.1 Liver dataset:

The liver dataset we downloaded as zip and it had extra files explaining the data. When we found the data it was called bupa.data we changed the name to liver.csv and added the column names by the order that the attributes appear in the table here.



Variables Table

Variable Name	Role	Type	Description	Units	Missing Values
mcv	Feature	Continuous	mean corpuscular volume		no
alkphos	Feature	Continuous	alkaline phosphotase		no
sgpt	Feature	Continuous	alanine aminotransferase		no
sgot	Feature	Continuous	aspartate aminotransferase		no
gammagt	Feature	Continuous	gamma-glutamyl transpeptidase		no
drinks	Target	Continuous	number of half-pint equivalents of alcoholic beverages drunk per day		no
selector	Other	Categorical	field created by the BUPA researchers to split the data into train/test sets		no

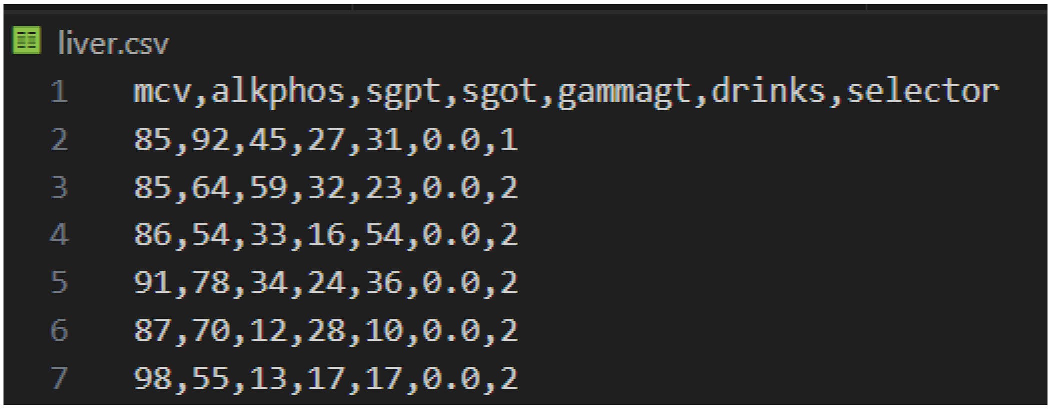
From the website.

- The first 5 variables are all blood tests which are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption.
- Each line in the dataset constitutes the record of a single male individual.
- The 7th field (selector) has been widely misinterpreted in the past as a dependent variable representing presence or absence of a liver disorder. This is incorrect. The 7th field was created by BUPA researchers as a train/test selector. It is not suitable as a dependent variable for classification. The dataset does not contain any variable representing presence or absence of a liver disorder.
- It recommended that if you want to use the dataset as a classification benchmark you should follow the method used in another article experiments by the donor *(Forsyth & Rada, 1986, Machine learning: applications in expert systems and information retrieval)* and others *(e.g. Turney, 1995, Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm)*, who used the 6th field (drinks), after dichotomising, as a dependent variable for classification.
  In the dataset documentation it said “It appears that drinks>5 is some sort of a selector on this database. See the PC/BEAGLE User's Guide for more information.”
  The PC/BEAGLE User's Guide" is a historical manual for data analysis software central to the early use of the UCI liver dataset.
  We did whoever drank more than 5 cups is considered sick (got a 1).

The article did a mistake and it used the selector as their target. We didn't follow the mistake.

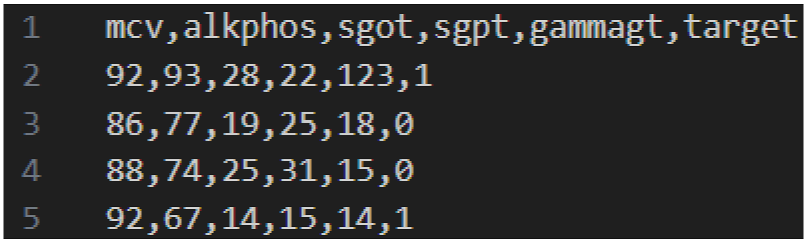
- We added a column “target”. And deleted the columns “drink” and “selector”.
- In the documentation of the dataset it stated that there are 4 duplicate rows. we delete them. So we started with 345 rows and after cleaning we have 341 rows.
- No missing values
- Because we were recreating the article we switched the order of the columns “sgpt” and “sgot” like they did in the article.
- We shuffled the dataset.

This is how the liver dataset looked before cleaning:



```
mcv,alkphos,sgpt,sgot,gammagt,drinks,selector
85,92,45,27,31,0.0,1
85,64,59,32,23,0.0,2
86,54,33,16,54,0.0,2
91,78,34,24,36,0.0,2
87,70,12,28,10,0.0,2
98,55,13,17,17,0.0,2
```

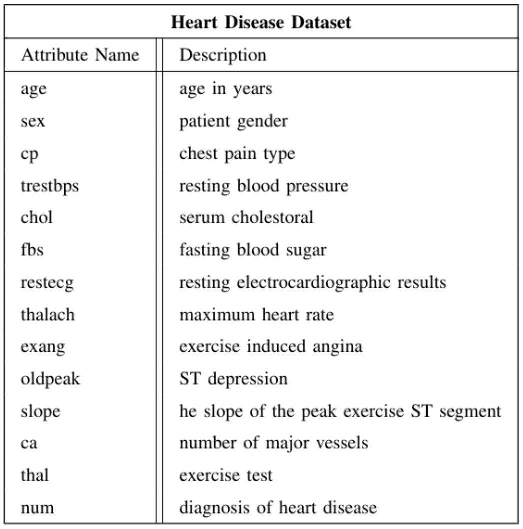
This is how it looked after cleaning the dataset:



```
mcv,alkphos,sgot,sgpt,gammagt,target
92,93,28,22,123,1
86,77,19,25,18,0
88,74,25,31,15,0
92,67,14,15,14,1
```

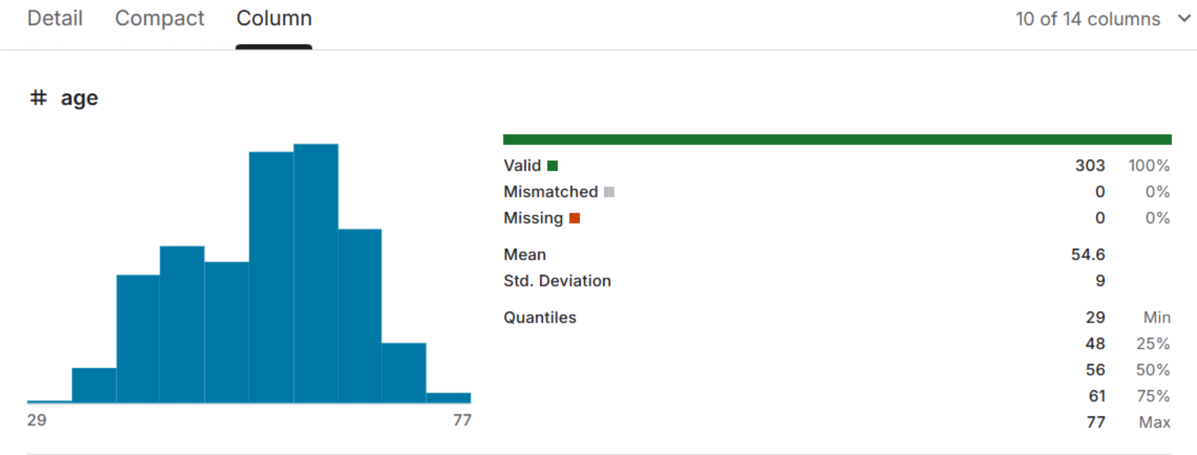
1.2 Heart dataset:

We found it in kaggle, we downloaded it as heart.csv and it already came with names of the columns.



Heart Disease Dataset	
Attribute Name	Description
age	age in years
sex	patient gender
cp	chest pain type
trestbps	resting blood pressure
chol	serum cholestoral
fbs	fasting blood sugar
restecg	resting electrocardiographic results
thalach	maximum heart rate
exang	exercise induced angina
oldpeak	ST depression
slope	he slope of the peak exercise ST segment
ca	number of major vessels
thal	exercise test
num	diagnosis of heart disease

- It has a graph like this for all columns and it shows that there is no missing data.

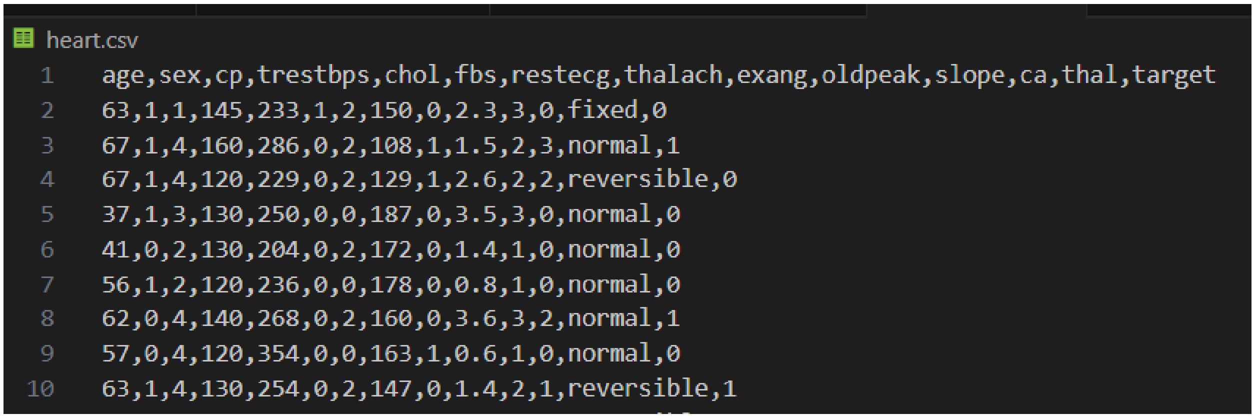


age

Valid 🟩	303	100%
Mismatched ⬜	0	0%
Missing 🟥	0	0%
Mean	54.6	
Std. Deviation	9	
Quantiles	29	Min
	48	25%
	56	50%
	61	75%
	77	Max

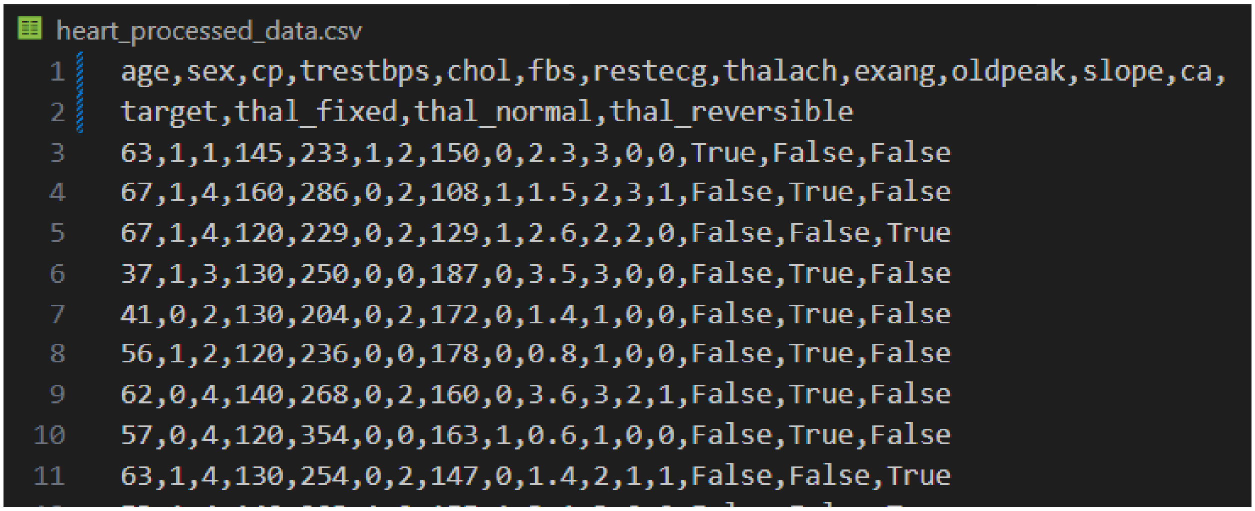
- 303 rows
- The thal column has categorial values normal, fixed and reversible so we did one hot encoding and added dummy columns. Thal_fixed, thal_normal, thal_reversible with true where it was. If it said normal we changed it to be true in thal_normal and false in the others.

  We cleaned the heart dataset in python.
  Before cleaning the dataset it looked like this:



```
age,sex,cp,trestbps,chol,fbs,restecg,thalach,exang,oldpeak,slope,ca,thal,target
63,1,1,145,233,1,2,150,0,2.3,3,0,fixed,0
67,1,4,160,286,0,2,108,1,1.5,2,3,normal,1
67,1,4,120,229,0,2,129,1,2.6,2,2,reversible,0
37,1,3,130,250,0,0,187,0,3.5,3,0,normal,0
41,0,2,130,204,0,2,172,0,1.4,1,0,normal,0
56,1,2,120,236,0,0,178,0,0.8,1,0,normal,0
62,0,4,140,268,0,2,160,0,3.6,3,2,normal,1
57,0,4,120,354,0,0,163,1,0.6,1,0,normal,0
63,1,4,130,254,0,2,147,0,1.4,2,1,reversible,1
```

After cleaning the dataset it looks like this:



```
age,sex,cp,trestbps,chol,fbs,restecg,thalach,exang,oldpeak,slope,ca,
target,thal_fixed,thal_normal,thal_reversible
63,1,1,145,233,1,2,150,0,2.3,3,0,0,True,False,False
67,1,4,160,286,0,2,108,1,1.5,2,3,1,False,True,False
67,1,4,120,229,0,2,129,1,2.6,2,2,0,False,False,True
37,1,3,130,250,0,0,187,0,3.5,3,0,0,False,True,False
41,0,2,130,204,0,2,172,0,1.4,1,0,0,False,True,False
56,1,2,120,236,0,0,178,0,0.8,1,0,0,False,True,False
62,0,4,140,268,0,2,160,0,3.6,3,2,1,False,True,False
57,0,4,120,354,0,0,163,1,0.6,1,0,0,False,True,False
63,1,4,130,254,0,2,147,0,1.4,2,1,1,False,False,True
```

2. Baseline Clustering:

Previous ways to cluster the dataset was to run traditional Kmeans and DBSCAN. They both have limitations, they are sensitive to hyperparameters.
We ran regular Kmeans and DBSCAN to see what we get and then to compare with the accuracy we get with our proposed algorithm.

2.1 Kmeans

we tried a few Ks.

```
set.seed(123)
k <- 5
kmeans_result <- kmeans(scaled_features, centers = k)
```

- Heart dataset:

k = 2 Weighted Classification Accuracy: 77.41 %

k = 3 Weighted Classification Accuracy: 85.38 %

k = 5 Weighted Classification Accuracy: 83.06 %
k = 10 Weighted Classification Accuracy: 82.39 %
k = 20 Weighted Classification Accuracy: 82.39 %

- Liver dataset:

k = 2 Weighted Classification Accuracy: 57.97 %
k = 5 Weighted Classification Accuracy: 57.97 %
k = 10 Weighted Classification Accuracy: 62.61 %
k = 15 Weighted Classification Accuracy: 62.03 %
k = 35 Weighted Classification Accuracy: 66.09 %
k = 40 Weighted Classification Accuracy: 69.86 %

Its a lot of guessing for the k… the new algorithm is supposed to calculate the optimal k for us.

2.2 DBSCAN

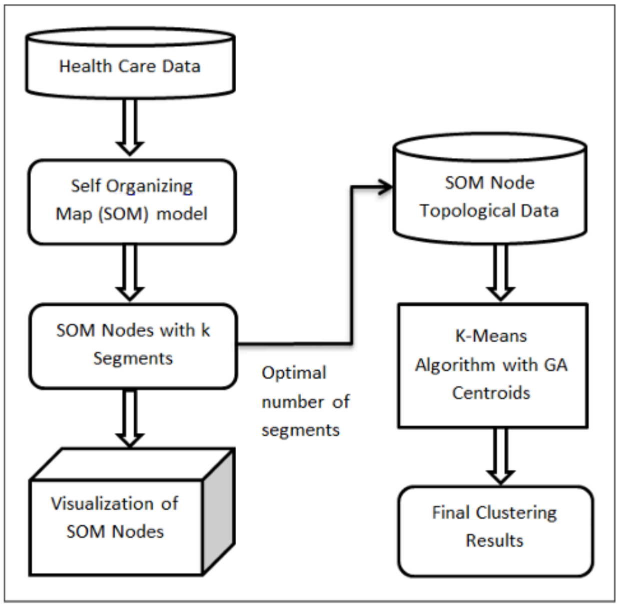
- Heart dataset:
  eps = 3, minPts = 1:   78.74%
  eps = 3.5, minPts = 2:   74.83%
  eps = 3.8, minPts = 3:   73.33%
  eps = 4, minPts = 4:   72.33%
  eps = 4.1, minPts = 8:   72.67%
  eps = 4.2, minPts = 16:   72.67%
  eps = 4.4, minPts = 20:   72.67%

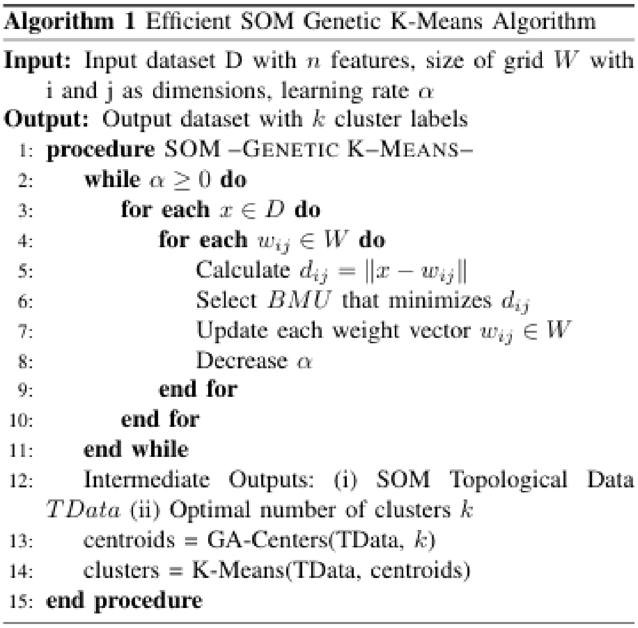
- Liver dataset:
  eps = 1.4, minPts = 1:   66.87%
  eps = 1.6, minPts = 2:   64.81%
  eps = 1.7, minPts = 3:   64.71%
  eps = 1.8, minPts = 5:   64.81%
  eps = 2, minPts = 7:   57.59%
  eps = 2.2, minPts = 12:   57.54%

To determine the optimal parameters for DBSCAN, I used the k-distance graph to identify the 'elbow' point, which provides the best estimate for the epsilon value. However, the clustering results are still not great.

3. Advanced clustering:

We used a combined clustering method that includes Self-Organizing Map (SOM), Genetic Algorithm (GA), and K-Means to analyze the datasets effectively.


Health Care Data
Self Organizing Map (SOM) model
SOM Nodes with k Segments
Visualization of SOM Nodes
Optimal number of segments
SOM Node Topological Data
K-Means Algorithm with GA Centroids
Final Clustering Results



Algorithm 1 Efficient SOM Genetic K-Means Algorithm

```
Input: Input dataset D with $n$ features, size of grid $W$ with
    i and j as dimensions, learning rate $\alpha$
Output: Output dataset with $k$ cluster labels
procedure SOM –Genetic K–Means–
    while $\alpha \geq 0$ do
        for each $x \in D$ do
            for each $w_{ij} \in W$ do
                Calculate $d_{ij} = \|x - w_{ij}\|$
                Select $BMU$ that minimizes $d_{ij}$
                Update each weight vector $w_{ij} \in W$
                Decrease $\alpha$
            end for
        end for
    end while
    Intermediate Outputs: (i) SOM Topological Data
  $TData$ (ii) Optimal number of clusters $k$
    centroids = GA-Centers(TData, $k$)
    clusters = K-Means(TData, centroids)
end procedure
```

*Because we did the target differently then the way they did in the article we got slightly different outputs.
*The complete code is in the git repository(link on page 1) in file som_genetic_kmeans.r

3.1 SOM

The process started with SOM, an unsupervised neural network that projects high-dimensional data onto a two-dimensional grid while keeping the relationships between data points. This helps group similar points close together, revealing natural clusters.
We chose a 6×6 grid (36 nodes) based on a rule suggested by Merényi et al. [33], which recommends that each neuron in the SOM represents about 10 data points to balance detail and generalization. Since our liver disease dataset has 345 records, dividing by 10 gives about 35 neurons. To keep the map consistent, we rounded to 36 neurons, which fits a 6×6 grid.
345/10 =~35 ->36(6x6) nodes
The original paper didn't clearly explain how to pick the number of clusters from the SOM, so we used a method that considers clusters with more than 5% of the data points as important to estimate the number of clusters *k*.

```
library(kohonen)    # SOM
```

```
som_grid <- somgrid(xdim = 6, ydim = 6, topo = "hexagonal")
set.seed(123)
som_model <- som(X_scaled, grid = som_grid, rlen = 750, alpha = c(0.5,
0.01))
# X_scaled- Normalize features (scale to mean=0 sd=1)
```

3.2 genetic algorithm

After deciding the number of clusters from the SOM, we applied a Genetic Algorithm to find the best starting points (centroids) for clustering. The GA used the SOM output as a starting guide and tested many combinations of cluster centers. It aimed to minimize the distance between data points and their nearest centers, helping to avoid poor starting points and bad clustering results.

```
library(GA)         # Genetic Algorithm
ga_result <- ga(
  type = "real-valued",
  fitness = fitness_function,
  lower = rep(apply(X_scaled, 2, min), k),
  upper = rep(apply(X_scaled, 2, max), k),
  popSize = population_size,
  maxiter = generations,
  run = 50,
  suggestions = suggestion
)
best_centroids <- matrix(ga_result@solution, nrow = k, byrow = TRUE)
```

3.2 K-means

With the number of clusters and starting centers set by the Genetic Algorithm, we ran K-Means to improve the cluster assignments. This combined method took advantage of SOM's ability to keep the data structure and help identify the appropriate number of clusters (*k*), GA's global search for good centers, and K-Means' ability to fine-tune the clusters.

The final results showed better accuracy and clearer cluster separation, especially when checked against known labels in the liver and heart disease datasets.

```
library(cluster)    # kmeans
kmeans_result <- kmeans(X_scaled, centers = best_centroids, iter.max =300)
# Best centroids- optimized centroids from GA
```

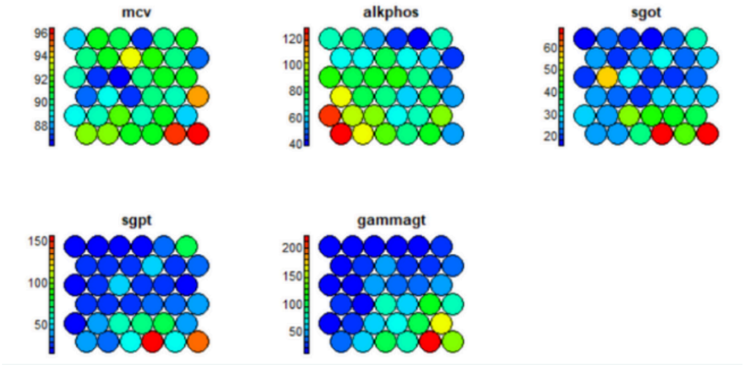
SOM-Genetic K-Means accuracy:

- Liver dataset: 77.42%
- Heart dataset: 83.39%

4. Evaluate performances:

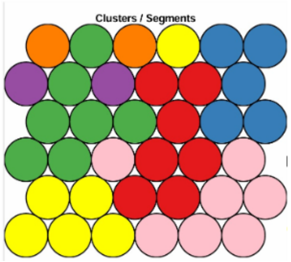
*The complete code is in the git repository(link on page 1) in file som_genetic_kmeans.r

4.1 Figure 2: Heatmap



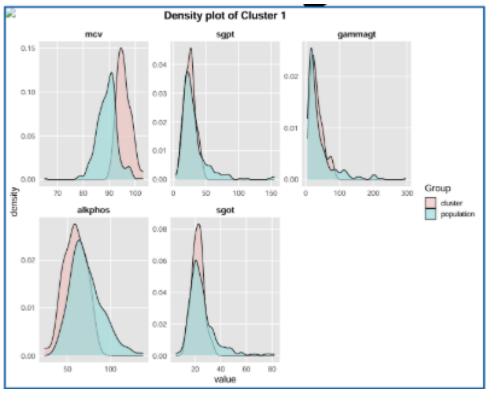
Shows heatmaps of the Self-Organizing Map (SOM) for each feature in the liver disease dataset. Each heatmap displays the scaled values of one variable, helping us see how features change across clusters. For example, the heatmaps reveal an inverse relationship between the variables alkphos and sgpt, where low values of one usually match high values of the other. These visualizations make it easier to understand the different characteristics and relationships between variables within the clusters identified by the SOM.

4.2 Figure 3: Clusters



Shows the final clustering result for the liver disease dataset using the SOM Genetic K-Means algorithm. The SOM first organized the data, followed by grouping into 7 clusters. Each color represents a cluster, highlighting how patient groups were separated. A similar process on the heart disease dataset resulted in 6 clusters.

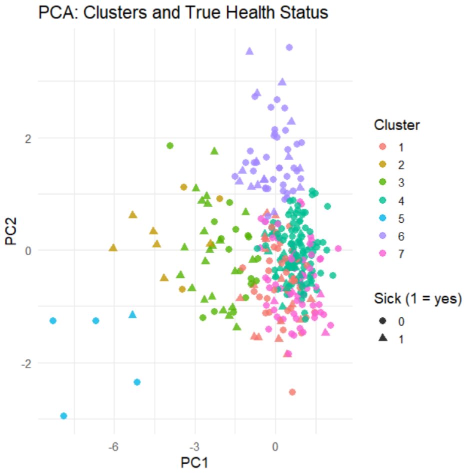
4.3 Figure 4: Density plot

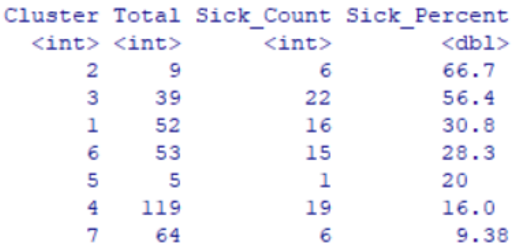


presents a density plot comparing cluster 1 to the rest of the population in the liver disease dataset.

4.4 results:

We went beyond the article, and we showed the visualization of the clusters. We can see here the representation of the liver dataset(with dimension reduction to 2D) if a patient is classified in cluster 2 then he is 66.7% sick. Where else if he is classified in cluster 7 then he has 9.37% sick, it is safe to say that he is healthy.

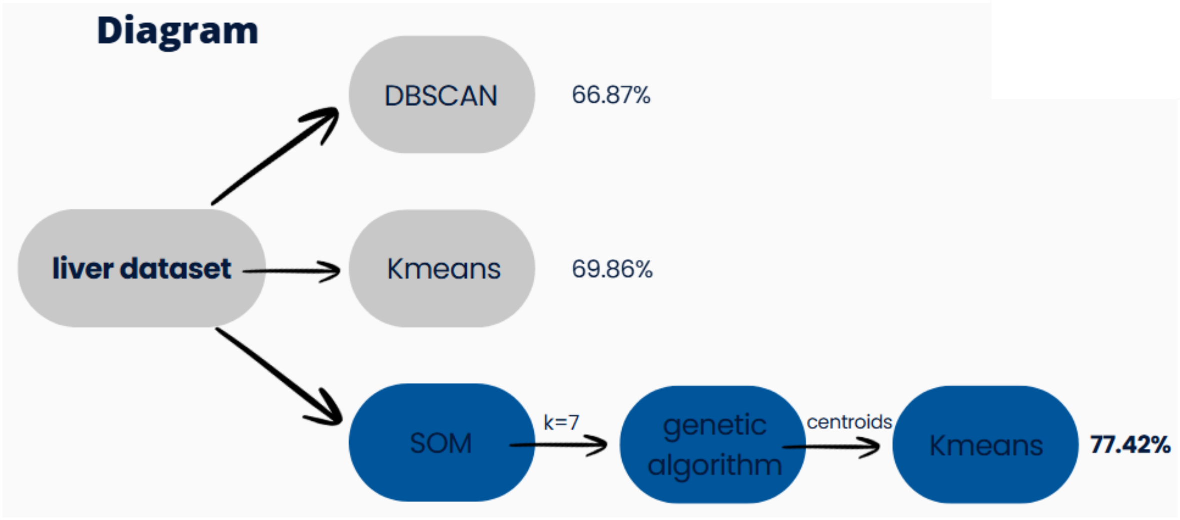




```
Cluster Total Sick_Count Sick_Percent
  <int> <int>      <int>        <dbl>
      2     9          6         66.7
      3    39         22         56.4
      1    52         16         30.8
      6    53         15         28.3
      5     5          1         20
      4   119         19         16.0
      7    64          6          9.38
```

Conclusion:

After running the proposed algorithm from the article and comparing the accuracy to the traditional Kmeans and DBSCAN the SOM-Genetic-Kmeans algorithm is more accurate, trust worthy and can help in the health world to detect and treat the patients more effectively.



Bibliography:

We used from the article

We used, new, not from the article

[1] Hian Chye Koh, Gerald Tan, et al. Data mining applications in healthcare. Journal of healthcare information management, 19(2):65, 2011.

[2] Mary K Obenshain. Application of data mining techniques to healthcare data. Infection Control & Hospital Epidemiology, 25(08):690–695, 2004.

[3] Neesha Jothi, Wahidah Husain, et al. Data mining in healthcare–a review. Procedia Computer Science, 72:306–313, 2015.

[4] Cristoph Helma, Tobias Cramer, Stefan Kramer, and Luc De Raedt. Data mining and machine learning techniques for the identification of mutagenicity inducing substructures and structure activity relationships of noncongeneric compounds. In J. Chem. Inf. Comput. Sci, volume 44, pages 1402–1411, 2004.

[5] Divya Tomar and Sonali Agarwal. A survey on data mining approaches for healthcare. International Journal of Bio-Science and Bio-Technology, 5(5):241–266, 2013.

[6] Brian Kulis and Michael I Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. arXiv preprint arXiv:1111.0352, 2011.

[7] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[8] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means

clustering algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1):100–108, 1979.

[9] Ramzi A. Haraty, Mohamad Dimishkieh, and Mehedi Masud. An enhancedk-means clustering algorithm for pattern discovery in health care data. International Journal of Distributed Sensor Networks, 2015:1 11, 2015.

[10] Taysir Hassan A. Soliman, Adel A. Sewissy, and Hisham AbdelLatif. A gene selection approach for classifying diseases based on microarray datasets. 2nd International Conference on Computer Technology and Development(ICCTD 2010), 2010.

[11] J. J. Tapia, Enrique Morett, and Edgar E. Vallejo. A clustering genetic algorithm for genomic data mining. Foundations of Computational Intelligence, 4:249–275, 2009.

[12] T. Kohonen. Self-Organizing Maps, volume 30. Springer, ISBN 3540679219, 2001.

[13] Ji-Jiang Yang, Jianqiang Li, Jacob Mulder, Yongcai Wang, Shi Chen, Hong Wu, Qing Wang, and Hui Pan. Emerging information technologies for enhanced healthcare. Computers in Industry, 69:3–11, 2015.

[14] Christo El Morr and Julien Subercaze. Knowledge management in healthcare. Handbook of research on developments in e-health and telemedicine: Technological and social perspetives, pages 490–510, 2010.

[15] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. AI magazine, 17(3):37, 1996.

[16] Rui Veloso, Filipe Portela, Manuel Filipe Santos, ´ Alvaro Silva, Fernando Rua, Ant´onio Abelha, and Jos´ e Machado. A clustering approach for predicting readmissions in intensive medicine. Procedia Technology, 16:1307–1316, 2014.

[17] Chao-Ton Su, Pa-Chun Wang, Yan-Cheng Chen, and Li-Fei Chen. Data mining techniques for assisting the diagnosis of pressure ulcer develop ment in surgical patients. Journal of medical systems, 36(4):2387–2399, 2012.

[18] Rub´en Arma˜ nanzas, Concha Bielza, Kallol Ray Chaudhuri, Pablo Martinez-Martin, and Pedro Larra˜naga. Unveiling relevant non-motor parkinson’s disease severity symptoms using a machine learning ap proach. Artificial intelligence in medicine, 58(3):195–202, 2013.

[19] Chih-Hung Jen, Chien-Chih Wang, Bernard C Jiang, Yan-Hua Chu, and Ming-Shu Chen. Application of classification techniques on develop ment an early-warning system for chronic illnesses. Expert Systems with Applications, 39(10):8852–8858, 2012.

[20] Pedro J Garc´ıa-Laencina, Pedro Henriques Abreu, Miguel Henriques Abreu, and No´emia Afonoso. Missing data imputation on the 5-year survival prediction of breast cancer patients with unknown discrete

values. Computers in biology and medicine, 59:125–133, 2015.

[21] Bichen Zheng, Sang Won Yoon, and Sarah S Lam. Breast cancer diagnosis based on feature extraction using a hybrid of k-means and support vector machine algorithms. Expert Systems with Applications, 41(4):1476–1482, 2014.

[22] Seokho Kang, Pilsung Kang, Taehoon Ko, Sungzoon Cho, Su-jin Rhee, and Kyung-Sang Yu. An efficient and effective ensemble of support vector machines for anti-diabetic drug failure prediction. Expert Systems with Applications, 42(9):4265–4273, 2015.

[23] Paul S Bradley, Usama M Fayyad, Cory Reina, et al. Scaling clustering algorithms to large databases. In KDD, pages 9–15, 1998.

[24] Charu C Aggarwal, Jiawei Han, Jianyong Wang, and Philip S Yu. A framework for clustering evolving data streams. In Proceedings of the 29th international conference on Very large data bases-Volume 29, pages 81–92. VLDB Endowment, 2003.

[25] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM SIAM symposium on Discrete algorithms, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[26] M Emre Celebi, Hassan A Kingravi, and Patricio A Vela. A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Systems with Applications, 40(1):200–210, 2013.

[27] Himanshu Gupta and Rajeev Srivastava. k-means based document clustering with automatic k selection and cluster refinement. International Journal of Computer Science and Mobile Applications, 2(5):7–13, 2014.

[28] Leonard Kaufman and Peter J Rousseeuw. Finding groups in data: an introduction to cluster analysis, volume 344. John Wiley & Sons, 2009.

[29] John Burkardt. K-means clustering. Virginia Tech, Advanced Research Computing, Interdisciplinary Center for Applied Mathematics, 2009.

[30] Ujjwal Maulik and Sanghamitra Bandyopadhyay. Genetic algorithm based clustering technique. Pattern recognition, 33(9):1455–1465, 2000.

[31] Martin Ester, Hans-Peter Kriegel, J¨org Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd, volume 96, pages 226–231, 1996.

[32] Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

[33] Erzs´ ebet Mer´enyi, Michael J Mendenhall, and Patrick ODriscoll. Advances in self-organizing maps and learning vector quantization

[34] Liver Disorders [Dataset]. (2016). UCI Machine Learning Repository. https://doi.org/10.24432/C54G67.