Experiment 05 - Clustering using RapidMiner Tool

Roll No.	19
Name	Manav Jawrani
Class	D15A
Subject	Business Intelligence Lab
LO Mapped	LO2: Organize and prepare the data needed for data mining algorithms in terms of attributes and class inputs, training, validating, and testing files. LO3: Implement the appropriate data mining methods like classification, clustering or association mining on large data sets using open source tools like WEKA
Grade	

DATE:

Aim - To implement the clustering algorithm using Rapidminer

Theory -

- K-means clustering -

1. K-means clustering is an unsupervised learning algorithm that partitions a set of data points into K clusters.

Algorithm -

1. The algorithm randomly selects K points from the data as initial centroids for each cluster.
2. Each data point is assigned to nearest centroid based on a distance metric, such as Euclidean distance.
3. The centroids of each cluster are updated as the mean of all data points in that cluster.
4. The algorithm iteratively assigns data points to clusters and updates centroids until the assignment of data points to cluster no longer changes, or a maximum no. of iterations is reached.
5. The final result is K clusters, where each data point is assigned to one of the K clusters and each cluster is represented by its centroid.

DATE:

Example -

Suppose that the data mining task is to cluster points (with (x,y) representing location) into three clusters, where the points are A1(2,10) A2(2,5), A3(8,4), B1(5,8), B2(7,5), B3(6,4) C1(1,2), C2(4,9). The distance function is Euclidean distance. Suppose initially we assign A1, B1 and C1 as the center of each cluster, respectively. Use the k-means algorithm to show only

- a. the three clusters center after the first round of execution.

→ Answer:

After the first round, the three new clusters are: 1 {A1}, 2 {B1, B2, A3, C2}. 3 {C1, A2} and their centers are 1(2,10), 2(6,6), 3(1.5, 3.5)

- The final three clusters.

→ Answer:

The Final clusters are 1 {A1, C2, B1}, 2 {A3, B2, B3}, 3 {C1, A2}.

DATE:

Hierarchical Clustering -

Hierarchical clustering is a technique used to group similar objects into clusters or groups based on their distance to each other. The method is called hierarchical because it creates a hierarchy of clusters that are nested within each other.

Steps -

1. Calculate pairwise distance between each pair of objects in dataset.
2. Represent each object as a single cluster.
3. Compute the distance between the closest clusters. This can be done using different linkage criteria such as single linkage.
4. Merge the two closest clusters into a single cluster, creating a new hierarchy.
5. Recalculate the distance between the new cluster and the remaining clusters.
6. Repeat steps 3-5 until all objects are in a single cluster.
7. Create a dendrogram then.

DATE:

- DBSCAN -

1. It is a density based clustering algorithm that groups together data points that are close to each other based on similarity.
2. The algorithm takes two parameters : 'epsilon' and 'min_samples', which define the radius of neighborhood around each data point and the minimum no. of points required to form cluster.
3. The steps of algorithm involve choosing an arbitrary point retrieving all points within 'epsilon' distance, determining whether the point is a core point or noise point, adding points to a cluster and repeating for all unvisited points.

- Observation -

1. K-means clustering -
   Avg. within centroid distance = -494.155
   Davies Bouldin = -0.831
2. Hierarchical clustering -
   No. of clusters = 10217
3. DBSCAN - No. of clusters = 57.

DATE:

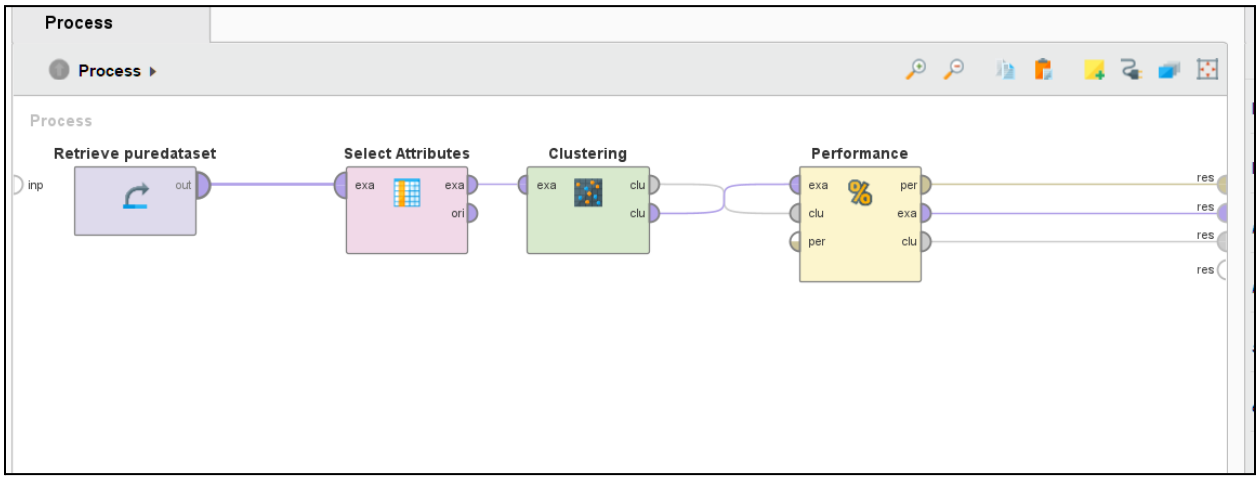
Conclusion -

1. K-means clustering : The avg. centroid distance = -494.155 indicating relatively well-separated clusters. Davies Bouldin index = -0.831 which is unusual and may indicate poor clustering.
2. Hierarchical clustering : No. of clusters = 10217 which is a very large numbes and may indicate overfitting of data.
3. DBSCAN clustering : No of clusters = 57 which is a more reasonable number. DBSCAN is a density based clustering algorithm.
4. Overall, K-means may not be good choice, hierarchical may be overfitting and DBSCAN clustering may be more reasonable choice.

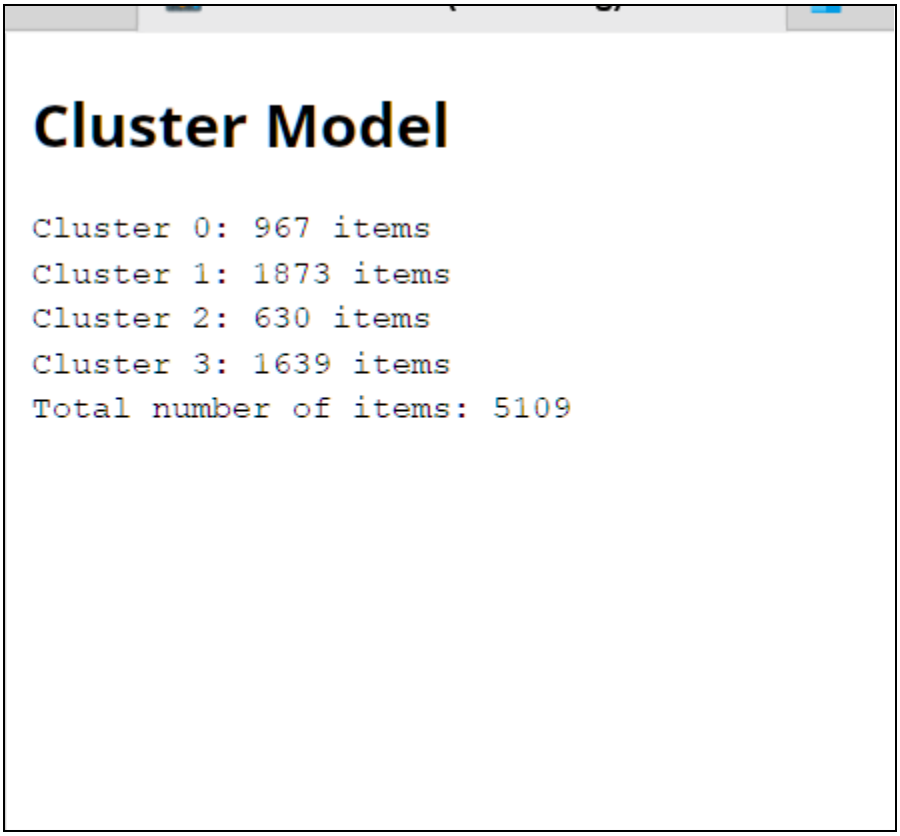
Implementation:

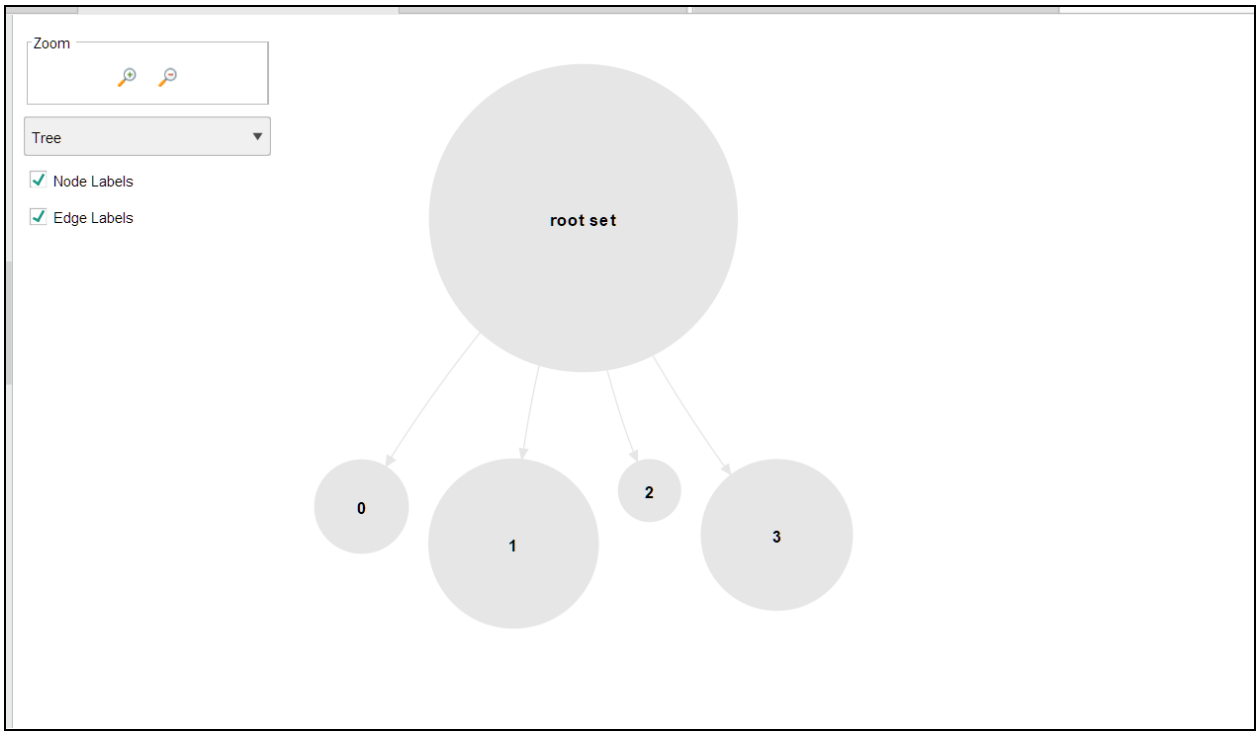
K-means clustering -

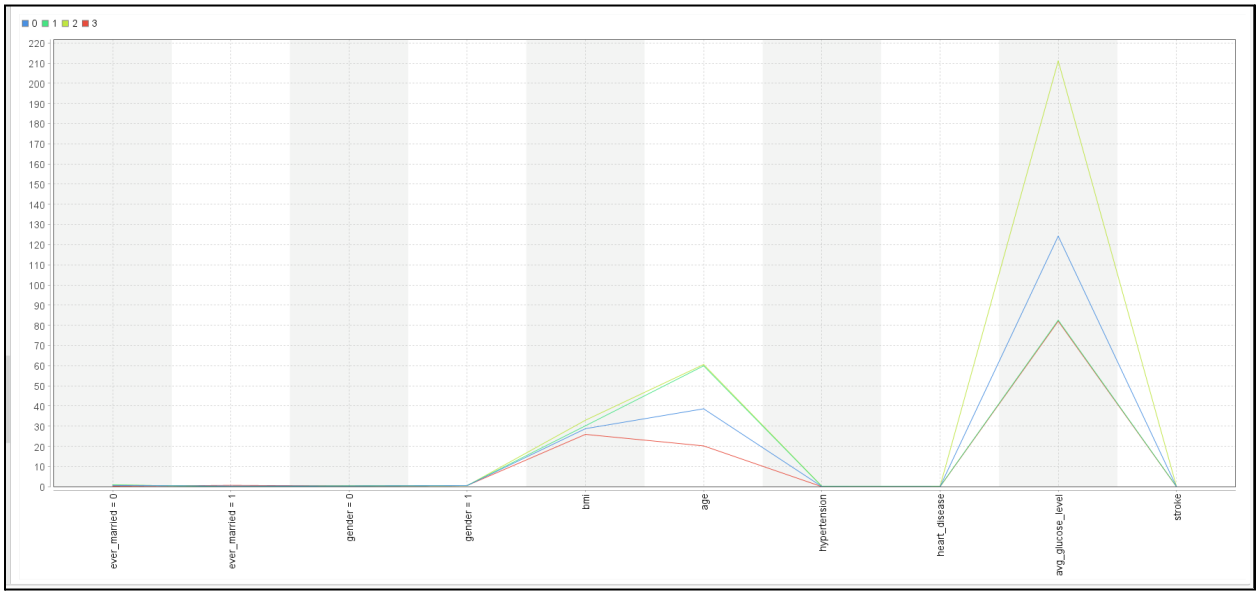
Process -



Output -




Zoom
Tree
Node Labels
Edge Labels
root set
0
1
2
3

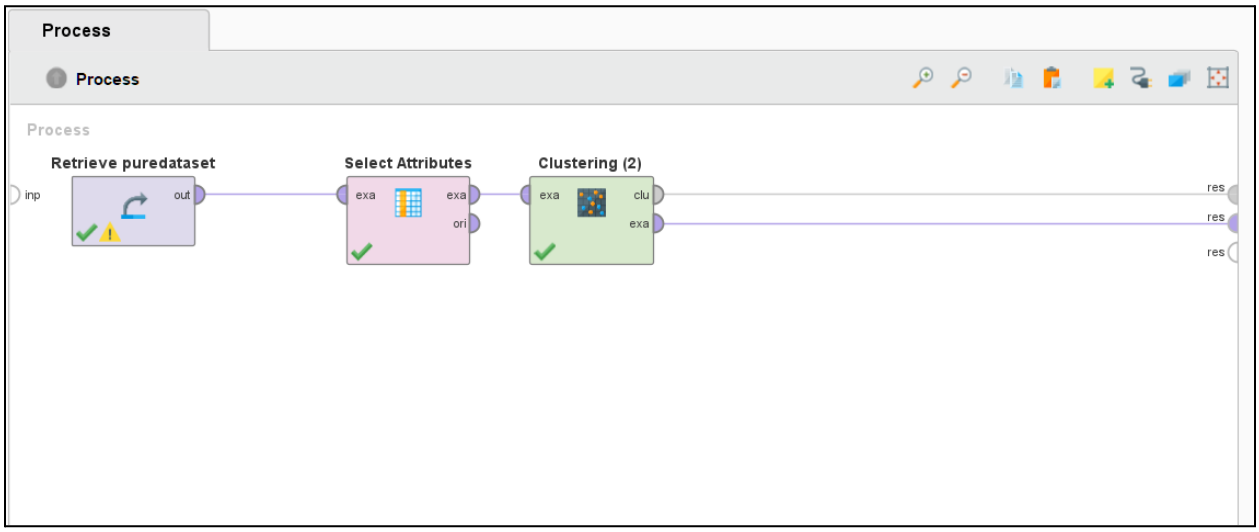

0 1 2 3
220
210
200
190
180
170
160
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
ever_married = 0
ever_married = 1
gender = 0
gender = 1
bmi
age
hypertension
heart_disease
avg_glucose_level
stroke

PerformanceVector

```
PerformanceVector:
Avg. within centroid distance: -494.155
Avg. within centroid distance_cluster_0: -692.945
Avg. within centroid distance_cluster_1: -393.160
Avg. within centroid distance_cluster_2: -751.926
Avg. within centroid distance_cluster_3: -393.202
Davies Bouldin: -0.831
```

Hierarchical clustering -

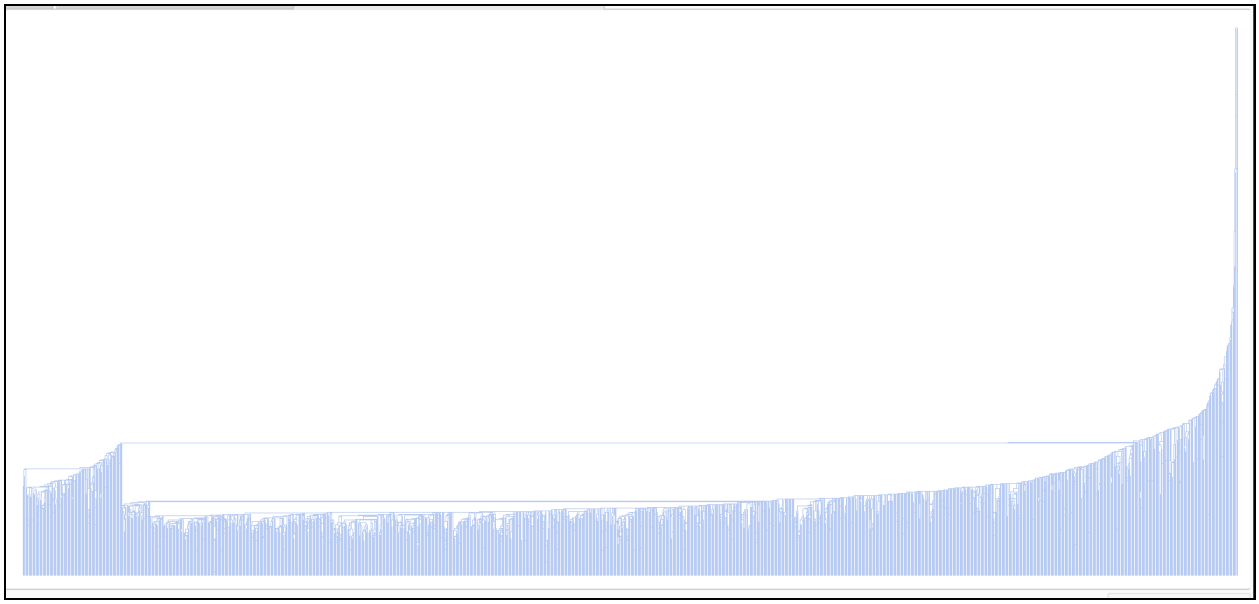
Process -



Output -

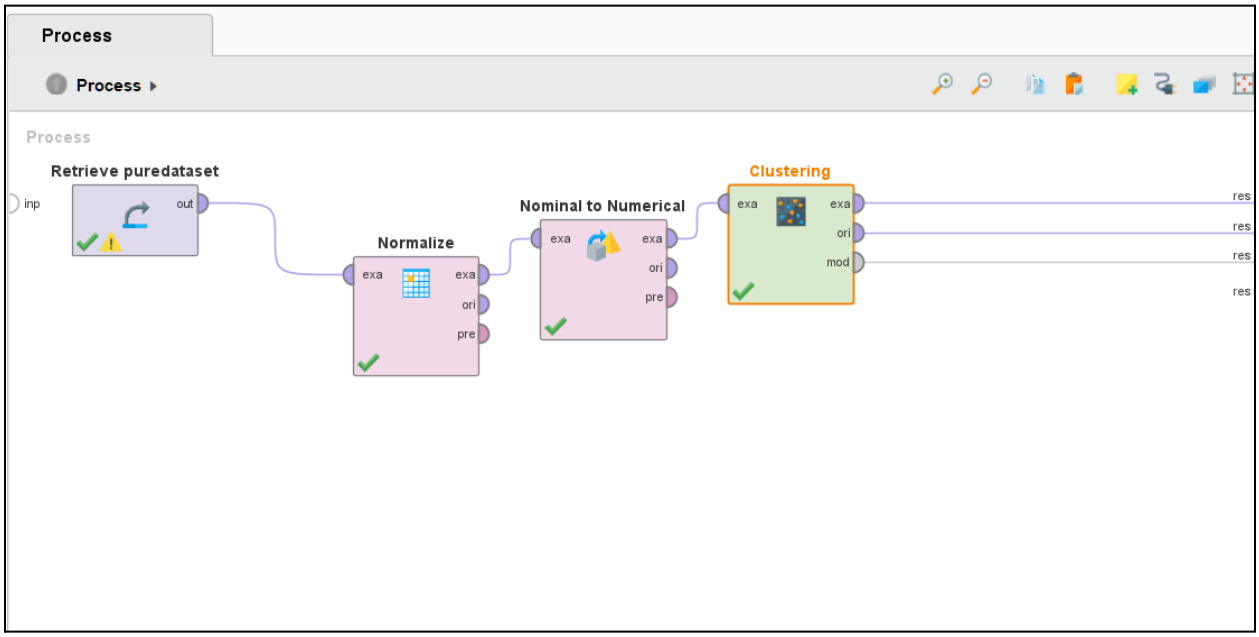
Hierarchical Cluster Model

```
Number of clusters :10217
Number of items :5109
```

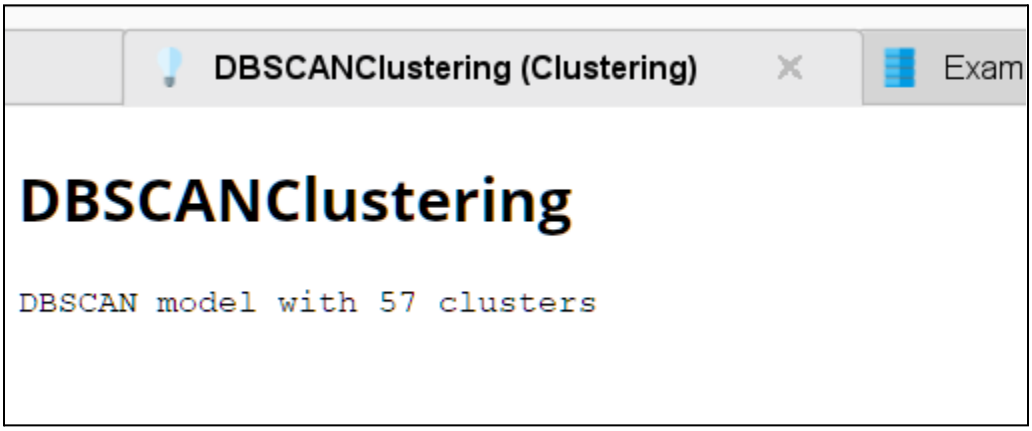


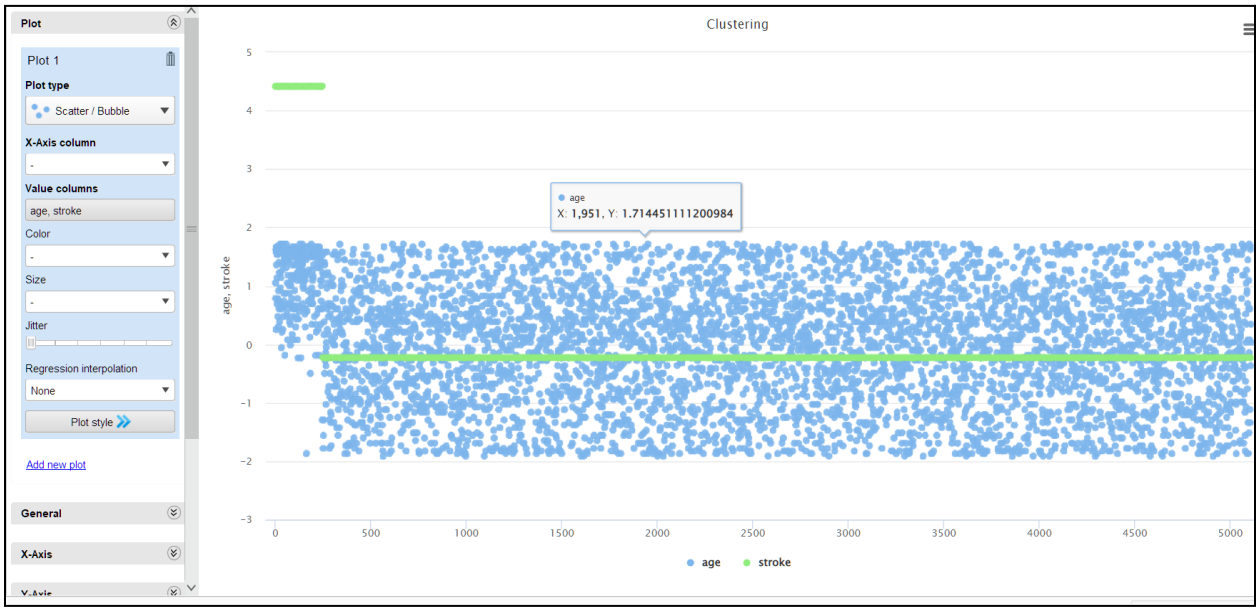
DBSCAN -

Process -



Output -


DBSCANClustering (Clustering)
Exam
DBSCANClustering
DBSCAN model with 57 clusters


Plot
Plot 1
Plot type
Scatter / Bubble
X-Axis column
-
Value columns
age, stroke
Color
-
Size
-
Jitter
Regression interpolation
None
Plot style
Add new plot
General
X-Axis
Clustering
age
X: 1,951, Y: 1.714451111200984
age, stroke
5
4
3
2
1
0
-1
-2
-3
0
500
1000
1500
2000
2500
3000
3500
4000
4500
5000
age
stroke