CS 422

DATA MINING

HOME WORK – 4

SUBMITTED BY :

SACHIN KRISHNA MURTHY

CWID : A20354077

Summary :

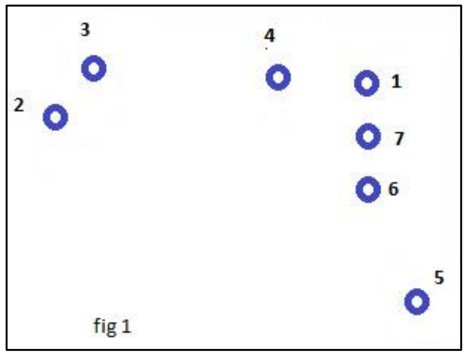
The following things has been accomplished and enclosed in this report :

- Solved all the questions given in part 1 along with suitable explanation.
- In Weka Part, for the data sets Iris and vote using SimpleKMeans and DBSCAN following this has been reported :
    - Attributes using preprocess and visualize tab, and analysis of it that impacts clustering.
    - Mentioned and explained the default parameters for each algorithm along with a description.
    - Reported the weka results for both the datasets using both the algorithms for the default values.
    - Experimented with varying the parameters and finding the best optimal solution and compared that with the results of the default parameters.
    - Ignored some different set of attributes of the datasets and analyzed the impacts of it on the performance of the classifier.
    - Finally concluded by analyzing the results obtained from the above experiments.
    - Also compared the performance of KMeans and DBSCAN on other data sets.

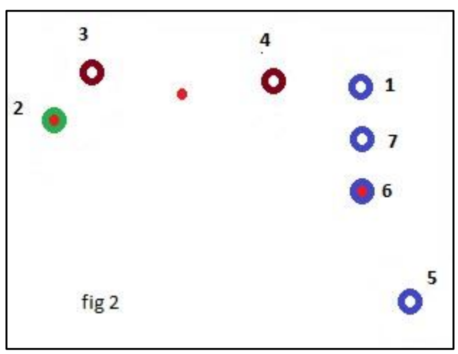
Part 1

1. **Find an example of a small set of points and three initial centroids so that kMeans with k=3 converges to a clustering with an empty cluster. Note that the initial centroids do not have to be members of the set of points. Explain your example in detail in your own words.**

Solution :



- Lets consider a data set with 7 points labelled 1-7 as shown in the above figure.
- Given K=3, hence we will divide these points into 3 clusters with centers or centroid as 2,3 and 5 as shown in fig.1 and run the first iteration.



After first iteration,

- 1,5,6 and 7 will be in one group say G1
- 3,4 will go to group, G2
- And 2 alone in a group, G3

After this process, new centroids are updated to new clusters as in fig 2, and second iteration is run.

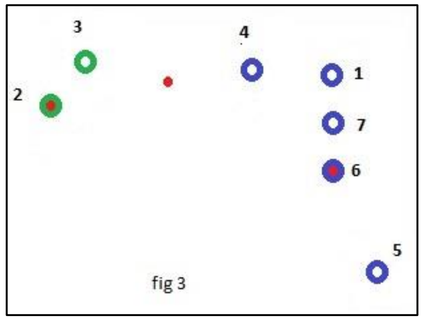


fig 3

After second iteration

- point 4 moves to G1, as it is nearer to centroid of G1 than that of G2
- point 3 moves towards G3 as it is close to centroid of G3

Now cluster G2 is empty as shown in fig3.

2. **We use SSE(RSS) as the measure of cluster quality and kMeans minimizes it. If there is an empty cluster, can that clustering be the global minimum solution based on RSS? Show all details of you arguments. Use your own words.**

<u>**Solution :**</u>

- No, if there is an empty cluster that clustering cannot be the global minimum solution based on RSS.
- SSE(sum of squared error) in the sum of squares of distance of each vector from a group.
- It can be given as :

$$\mathrm{RSS}_k = \sum_{\vec{x} \in \omega_k} |\vec{x} - \vec{\mu}(\omega_k)|^2$$

$$\mathrm{RSS} = \sum_{k=1}^{K} \mathrm{RSS}_k$$

- When there is an empty cluster its SSE will be larger. To reduce this SSE a replacement centroid is needed and this reduces the total SSE.
- The newly chosen centroid can be from the cluster farthest from this cluster and this will eliminate the point that gives most to the total SSE. This new centroid can also be chosen from a cluster with highest SSE. This will split the cluster and reduce the total SSE

- The empty cluster doesn't have any data but will have a larger SSE. Thus this cluster cannot be the global minimum solution for clustering. But this cluster can be used to reduce the overall SSE by choosing an alternate centroid form another suitable cluster.

3. **kMeans with soft cluster assignment**
   Computes the fractional membership of a document in a cluster as a function of the distance D from its centroid. That function is monotonically decreasing, e.g., as e^(1/d)
   Wrote very detailed pseudocode of kMeans using this soft version. Provide clear comments for each line of code, in your own words.

   Solution :

   The following steps has been executed in the below pseudocode :

   - Initially the whole dataset is divided into k clusters where k is predefined.
   - Then for these k clusters, centroids are found by getting an average of points in the cluster.
   - If a cluster is empty, a farthest point is considered as a centroid.
   - For all the points, distance with each centroid is found, and points are moved to cluster with whose centroid distance is least forming k new clusters.
   - Steps 2 to 5 is repeated until maxIterations.
   - maxdist is just a parameter to get minimum distance between a point and centroids.

```
max dist = 0;

//getting maxdist as one unit more than largest distance between any
//two points in the dataset
for each point i in dataset
                for each point j in dataset
                 if maxdist < |point i - point j|
                                maxdist = |point i - point j|
                end for
end for
maxdist + 1;
iteration = 0;

//actual k means begin here
divide data set into k random clusters
do{
       empty centroid list
                 for each cluster i = 0 to k-1
                    sum = 0;

                    for each point in ith cluster
                     sum = sum + point
                    end for

// getting cluster centroid as mean of cluster
                    if( cluster not empty)
                                centroid i = sum/number of points
```

```
          else
// handling empty cluster assigning farthest point
               dist = 0;
               for each point in dataset
                    if(dist <= |centroid i – point|)

// getting  farthest point
                              dist = |centroid i – point|
                              val = point
                    end if
               end for
               centroid i = val
          end if
          add centroid i to centroid list
     end for

//running iteration for re clustering
     for each point j in dataset
               pointdist = maxdist

               for each centroid i in centroid list
                         if( pointdist >= |centroid i – point j|)
                                   pointdist = |centroid i – point j|
                                   centVal = i;
                         end if

// moving point to  cluster of  nearest centroid
                         move point j to cluster centVal
               end for
     end for

     iteration++;
// stopping condition
while(iteration < maxIteration)}
```

<u>Exercise 7.4.1</u> : Consider two clusters that are a circle and a surrounding ring, as in the running example of this section. Suppose:

i. The radius of the circle is c.

ii. The inner and outer circles forming the ring have radii i and o, respectively.

iii. All representative points for the two clusters are on the boundaries of the clusters.

iv. Representative points are moved 20% of the distance from their initial position toward the centroid of their cluster.

v. Clusters are merged if, after repositioning, there are representative points from the two clusters at distance d or less.

In terms of d, c, i, and o, under what circumstances will the ring and circle be merged into a single cluster?

Solution :

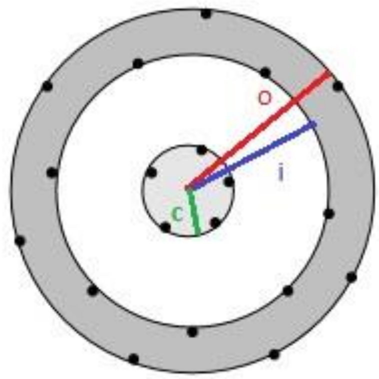


fig1

Before

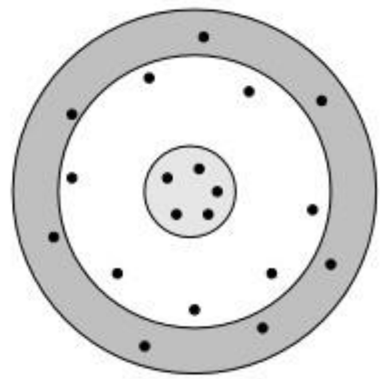


fig 2

After

As shown in fig.1 :

- The circle radius = c.
- The inner ring radius = i
- The outer ring radius = o

In such cases if the points are on the outer ring there are chances that they might move to cluster outside the cluster in fig.1. Therefore to avoid this a 20% repositioning of points is done as shown in fig.2.

Here Clusters are merged if there are representative points from the two clusters at distance d or less.

Clusters are merged if there are representative points from the two clusters at :

o - i <= d and **i-c <= d**

In this case all the points tend to move towards those on inner ring into a single cluster.

As the iteration repeats, points are moved towards the centroid by 20% and stops when no more are sufficiently close to cluster.

Weka Experiments

Datasets Used : Iris and Vote

Algorithms Used : SimpleKMeans and DBSCAN

SimpleKMeans Algorithm : The main objective of this algorithm is to classify and group similar objects by considering attributes or features which is done by minimizing the SSE between the object (data) and the centroid. In other words this algorithm attempts to find a user specified number of clusters (K) which are represented by the centroids.

- In the first step we randomly select K initial centroids, where K is the number of clusters which is user specified.
- Then determine the distance between the data point and the cluster center through calculation.
- Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.
- Thereafter we need to determine the new cluster center and also the distance between the data points and the new obtained cluster centers.
- We repeat these steps until the centroids remain the same.

DBSCAN Algorithm : This is a density based clustering algorithm which produces a partitional clustering. Here the number of clusters is automatically determined by the algorithm. This algorithm does not produce a complete clustering because points in low density regions are classified as noise and omitted. But it can easily identify clusters in huge spatial datasets by observing the local density of the elements.

- Initially all the points are labelled as core, border or noise points.
- In the next step noise points are eliminated.
- An edge is put between all core points that are within the epsilon of each other.
- Make each group of connected core points into a separate cluster.
- Assign each border point to one of the clusters of its associated core points.

1. **Use the preprocessing tab and the visualize tab to explore and visualize the attributes. Analyze what you see and how you think it can affect clustering.**

   Solution :

Dataset : Iris

- **Attributes :**
- **Using Preprocess tab :**

- Number of Attributes : 5

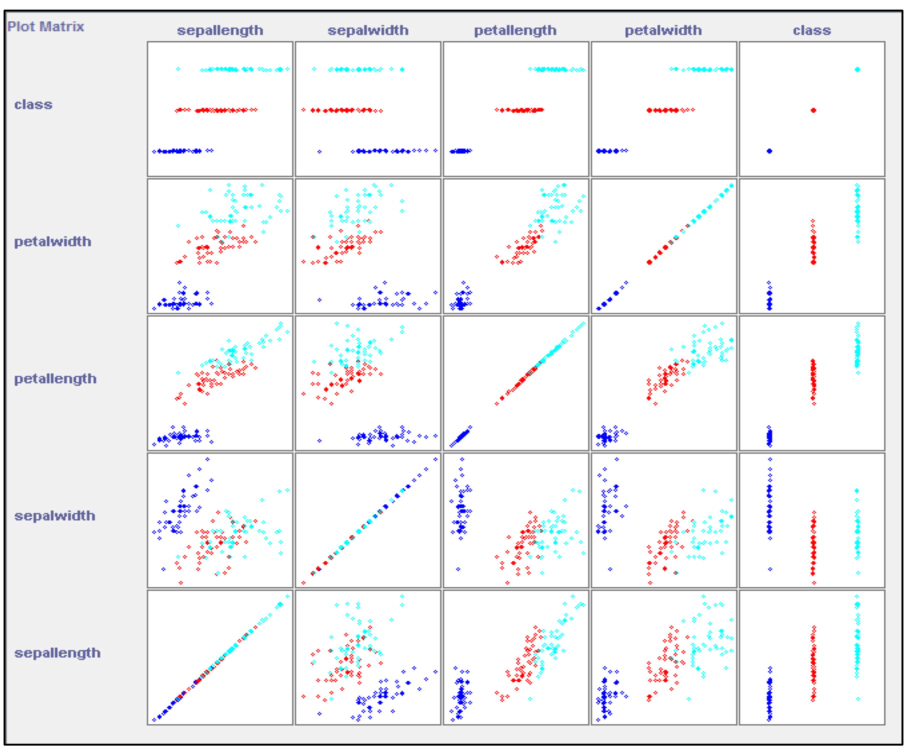
➢ Minimum, Maximum, Mean and Standard Deviation of the Attributes :

Sl No.	Attribute	Minimum Value	Maximum Value	Mean	Standard Deviation
1.	Sepal Length	4.3	7.9	5.843	0.828
2.	Sepal Width	2	4.4	3.054	0.434
3.	Petal Length	1	6.9	3.759	1.764
4.	Petal Width	0.1	2.5	1.199	0.763
5.	Class				

➢ Class Attribute :

Sl No.	Label	Count
1.	Iris-setosa	50
2.	Iris-versicolor	50
3.	Iris-virginica	50

➢ Using Visualize Tab :



- **Analysis :**

- The attribute values can be used to find the minimum distance from the centroid.
- From the preprocess tab and visualize tab we can observe the isolation of different classes in terms of their attribute values.
- In case of attributes petal length and petal width we can notice that Iris Setosa is isolated from other two classes when the values of these two attributes is low.
- As the value for attributes petal length and petal width goes on increasing the class versilcolor is visible and when the values for these attributes still increases further class Iris virginica can also be seen.
- From these observations from Visualize tab we can say that for attributes petal length and petal width Iris Sesota might form a different cluster.
- Also we can observe that the attributes Sepal length and sepal width are not clearly separated in the range of values.
- Hence excluding the attributes sepal length and sepal width might give better optimal solution.

- **Dataset : Vote**

- **Attributes :**

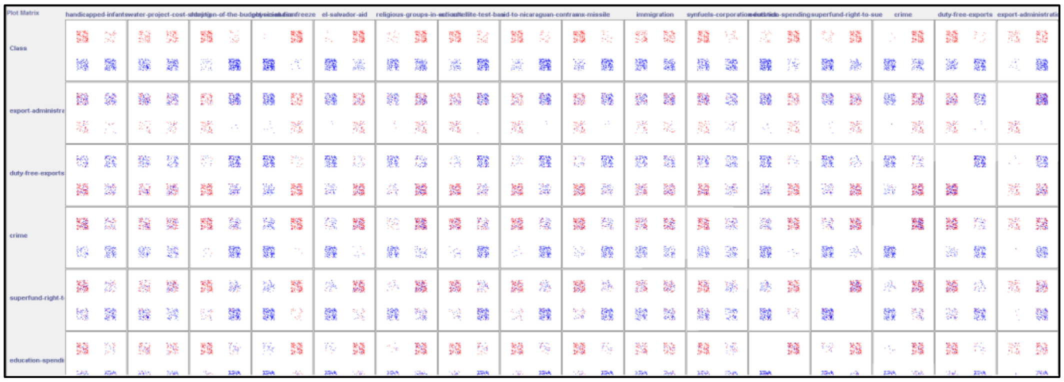
- **Using Preprocess Tab :**

- Number of Attributes : 17

Sl No.	Attribute Name	Label	Count
1.	Handicapped-infants	n	236
		y	187
2.	Water-project-cost-sharing	n	192
		y	195
3.	Adoption-of-the-budget-resolution	n	171
		y	253
4.	Physician-fee-freeze	n	247
		y	177
5.	El-salvador-aid	n	208
		y	212
6.	Religious-groups-in-schools	n	152
		y	272

7.	Anti-satellite-test-ban	n	182
		y	239
8.	Aid-to-nicaraquan-contras	n	178
		y	242
9.	mx-missile	n	206
		y	207
10.	Immigration	n	212
		y	216
11.	Synfuels-corporation-cutback	n	264
		y	150
12.	Education-spending	n	233
		y	171
13.	Superfund-right-to-sue	n	201
		y	209
14.	Crime	n	170
		y	248
15.	Duty-free-exports	n	233
		y	174
16.	Export-administration-act-south-africa	n	62
		y	269
17.	Class	democrat	267
		republican	168

➢ **<u>Using Visualize Tab :</u>**



➢ **<u>Analysis :</u>**

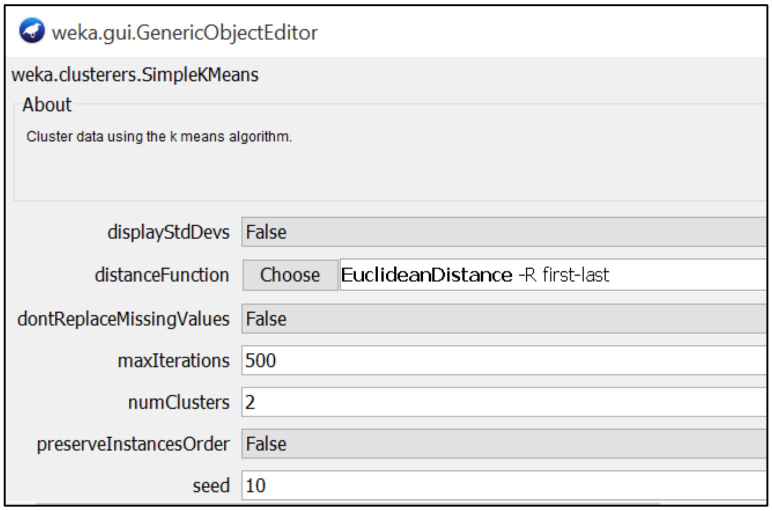
- Vote dataset has many attributes to consider.
- For the two attributes Physician-fee-freeze and El-salvador-aid we can notice that most of the data with value ‘yes’ for these two attributes are democrats and with value ‘no’ are republican.
- For the rest of other attributes the values are distributed without getting oriented to single class.

2. Cluster the data, use the default values, report the results for both algorithms. Explain how you evaluate the cluster performance in Weka, what measures do you see in the output tab, what do they mean.
Explain all parameters for each algorithm. What parameters should be changed for the experimentations and what parameters should be used in the default values and why.

<u>Solution :</u>

➢ **<u>SimpleKMeans Algorithm :</u>**

- **<u>Parameters and their default Values :</u>**



Parameter	Value
displayStdDevs	False
distanceFunction	Euclidian Distance
dontReplaceMissingValues	False
maxIterations	500
numClusters	2
preserveInstancesOrder	False
Seed	10

- **<u>Description Of Parameters :</u>**

- **displayStdDevs :** It displays the standard deviations of numeric attributes and counts of nominal attributes.

- **distanceFunction :** Here we can use 2 options :

    - Eucledian Distance
    - Manhattan Distance

  The distance function to use for instances comparison (default: weka.core.EuclideanDistance).

    - In Eucledian Distance, the function which computes the distance between two points is chosen.
    - While in case of Manhattan distance, the centroids are computed as the component-wise median instead of taking the mean.

- **dontReplaceMissingValues :** Replace missing values globally with mean/mode.
  This parameter has 2 options :
  - True : It does not replace the missing values with the mean/mode.
  - False : It replaces the missing values

- **maxIterations :** It sets the maximum number of iterations, that is the threshold until which it can compute the clusters.

- **numClusters :** It is the set of number of clusters which is the K value.

- **preserveInstancesOrder :** It preserves the order of instances in which they are taken.

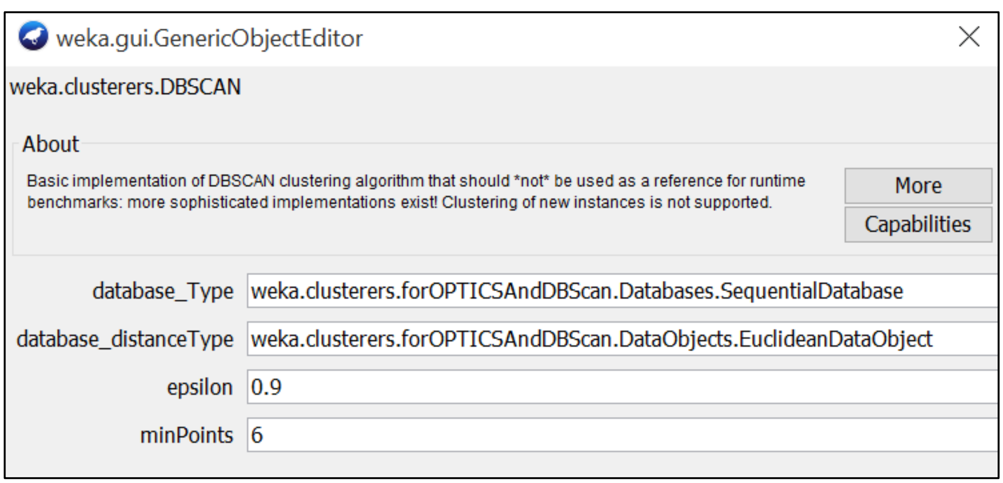
- **seed :** It is the random number to be used as seed. It gives us the better start and also helps in improving the performance of the algorithm by taking random centroids and later merging them.

- **Incorrectly Clustered Instances** : It gives us the count and percentage of the number of clusters that is mistakenly classified.

- **Sum of Squared Error (SSE) :** It is the sum of the square of the residuals i,e the deviations predicted from the actual empirical value of data. It also gives us the distance to the nearest cluster. We can fetch the SSE value by squaring these errors and adding them.

➢ **DBSCAN Algorithm :**

- **Parameters and their Default Values :**



Parameter	Value
database_Type	weka.clusterers.forOPTICSAndDBScan.Databases.SequentialDatabase
database_distanceType	weka.clusterers.forOPTICSAndDBScan.DataObjects.EuclideanDataObject
epsilon	0.9
minPoints	6

➢ **Description of Parameters :**

- **database_Type :** It delineates about the database used. Here we are using sequential database as default.

- **database_distanceType :** It delineates about the distance-type used. Here we are using Eucledian distance as default.

- **epsilon :** It is the radius of the epsilon-range-queries i,e radius with which the cluster is formed. Here we are using a default epsilon value of 0.9.

- **minPoints :** It is the minimun number of DataObjects required in an epsilon-range-query i,e the minimum number of points which needs to satisfied for the cluster to be formed.

➢ **Dataset : Iris**

➢ **Using SimpleKMeans Algorithm :**

- **KMeans Clustering Results :**

```
kMeans
======

Number of iterations: 7
Within cluster sum of squared errors: 12.143688281579722
Missing values globally replaced with mean/mode

Cluster centroids:
                          Cluster#
Attribute      Full Data         0          1
                   (150)     (100)       (50)
==============================================
sepallength       5.8433     6.262      5.006
sepalwidth         3.054     2.872      3.418
petallength       3.7587     4.906      1.464
petalwidth        1.1987     1.676      0.244
```

- **<u>Model and Evaluation on Training Set :</u>**

```
=== Model and evaluation on training set ===

Clustered Instances

0      100 ( 67%)
1       50 ( 33%)


Class attribute: class
Classes to Clusters:

  0  1  <-- assigned to cluster
  0 50 | Iris-setosa
 50  0 | Iris-versicolor
 50  0 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa

Incorrectly clustered instances :       50.0     33.3333 %
```

➢ **<u>Observations :</u>**

- In this case as the default for K or number of clusters is 2, we can notice 2 clusters.
- Cluster 1 has only one class sample i,e Iris setosa.
- The other cluster has both Iris versicolor and Iris virginica in it.
- For a cluster only one class label can be assigned.
- In cluster 0, since 2 classes are present only one class is taken to consideration and the other class is classified as incorrectly clustered instance.
- There are 50 incorrectly instances of Iris Virginica that is equal to 33.333% and SSE is 12.1436.

➢ **<u>Dataset : Iris</u>**

➢ **<u>Using DBSCAN Algorithm :</u>**

- **<u>DBSCAN Clustering Results :</u>**

```
DBSCAN clustering results
========================================================================================

Clustered DataObjects: 150
Number of attributes: 4
Epsilon: 0.9; minPoints: 6
Index: weka.clusterers.forOPTICSAndDBScan.Databases.SequentialDatabase
Distance-type: weka.clusterers.forOPTICSAndDBScan.DataObjects.EuclideanDataObject
Number of generated clusters: 1
```

- **Model and Evaluation on training Set :**

```
=== Model and evaluation on training set ===

Clustered Instances

0      150 (100%)


Class attribute: class
Classes to Clusters:

  0  <-- assigned to cluster
 50 | Iris-setosa
 50 | Iris-versicolor
 50 | Iris-virginica

Cluster 0 <-- Iris-setosa

Incorrectly clustered instances :       100.0    66.6667 %
```

```
(  0.) 5.1,3.5,1.4,0.2                                    -->  0
(  1.) 4.9,3,1.4,0.2                                      -->  0
(  2.) 4.7,3.2,1.3,0.2                                    -->  0
(  3.) 4.6,3.1,1.5,0.2                                    -->  0
(  4.) 5,3.6,1.4,0.2                                      -->  0
(  5.) 5.4,3.9,1.7,0.4                                    -->  0
(  6.) 4.6,3.4,1.4,0.3                                    -->  0
(  7.) 5,3.4,1.5,0.2                                      -->  0
(  8.) 4.4,2.9,1.4,0.2                                    -->  0
(  9.) 4.9,3.1,1.5,0.1                                    -->  0
( 10.) 5.4,3.7,1.5,0.2                                    -->  0
( 11.) 4.8,3.4,1.6,0.2                                    -->  0
( 12.) 4.8,3,1.4,0.1                                      -->  0
( 13.) 4.3,3,1.1,0.1                                      -->  0
( 14.) 5.8,4,1.2,0.2                                      -->  0
( 15.) 5.7,4.4,1.5,0.4                                    -->  0
```

.
.
.
.

```
(125.) 7.2,3.2,6,1.8                                      -->  0
(126.) 6.2,2.8,4.8,1.8                                    -->  0
(127.) 6.1,3,4.9,1.8                                      -->  0
(128.) 6.4,2.8,5.6,2.1                                    -->  0
(129.) 7.2,3,5.8,1.6                                      -->  0
(130.) 7.4,2.8,6.1,1.9                                    -->  0
(131.) 7.9,3.8,6.4,2                                      -->  0
(132.) 6.4,2.8,5.6,2.2                                    -->  0
(133.) 6.3,2.8,5.1,1.5                                    -->  0
(134.) 6.1,2.6,5.6,1.4                                    -->  0
(135.) 7.7,3,6.1,2.3                                      -->  0
(136.) 6.3,3.4,5.6,2.4                                    -->  0
(137.) 6.4,3.1,5.5,1.8                                    -->  0
(138.) 6,3,4.8,1.8                                        -->  0
(139.) 6.9,3.1,5.4,2.1                                    -->  0
(140.) 6.7,3.1,5.6,2.4                                    -->  0
(141.) 6.9,3.1,5.1,2.3                                    -->  0
(142.) 5.8,2.7,5.1,1.9                                    -->  0
(143.) 6.8,3.2,5.9,2.3                                    -->  0
(144.) 6.7,3.3,5.7,2.5                                    -->  0
(145.) 6.7,3,5.2,2.3                                      -->  0
(146.) 6.3,2.5,5,1.9                                      -->  0
(147.) 6.5,3,5.2,2                                        -->  0
(148.) 6.2,3.4,5.4,2.3                                    -->  0
(149.) 5.9,3,5.1,1.8                                      -->  0
```

- **Observations :**

- Here we can observe the formation of only one cluster.
- Also all the three classes are in the same cluster.
- We know that only a single class can be assigned to each cluster. Here Iris setosa is assigned to the cluster as it comes first.
- The other two classes are said to be incorrectly classified.
- There are 100 incorrectly classified instances which is equal to 66.667%

- **Dataset : Vote**

- **Using SimpleKmeans Algorithm :**

- **KMeans Clustering Results :**

```
kMeans
======

Number of iterations: 3
Within cluster sum of squared errors: 1449.0
Missing values globally replaced with mean/mode

Cluster centroids:
                                                     Cluster#
Attribute                                Full Data          0          1
                                             (435)      (207)      (228)
=========================================================================
handicapped-infants                              n          n          y
water-project-cost-sharing                       y          y          n
adoption-of-the-budget-resolution                y          n          y
physician-fee-freeze                             n          y          n
el-salvador-aid                                  y          y          n
religious-groups-in-schools                      y          y          n
anti-satellite-test-ban                          y          n          y
aid-to-nicaraguan-contras                        y          n          y
mx-missile                                       y          n          y
immigration                                      y          y          y
synfuels-corporation-cutback                     n          n          n
education-spending                               n          y          n
superfund-right-to-sue                           y          y          n
crime                                            y          y          n
duty-free-exports                                n          n          y
export-administration-act-south-africa           y          y          y
```

- **Model and Evaluation On Training Set :**

```
=== Model and evaluation on training set ===

Clustered Instances

0      207 ( 48%)
1      228 ( 52%)


Class attribute: Class
Classes to Clusters:

   0   1  <-- assigned to cluster
  50 217 | democrat
 157  11 | republican

Cluster 0 <-- republican
Cluster 1 <-- democrat

Incorrectly clustered instances :       61.0     14.023  %
```

➢ **Observations :**

- In this case only 2 clusters are formed.
- Cluster 0 is assigned to republican as it has higher number of republican instances and cluster 1 is assigned to democrat.
- There are 50 + 11 = 61 incorrectly classified instances which is equal to 14.023% and the SSE is 1449.

➢ **Dataset : Vote**

➢ **Using DBSCAN Algorithm :**

- **DBSCAN Clustering Results :**

```
DBSCAN clustering results
========================================================================================

Clustered DataObjects: 435
Number of attributes: 16
Epsilon: 0.9; minPoints: 6
Index: weka.clusterers.forOPTICSAndDBScan.Databases.SequentialDatabase
Distance-type: weka.clusterers.forOPTICSAndDBScan.DataObjects.EuclideanDataObject
Number of generated clusters: 14
```

- **Model and Evaluation on Training Set :**

```
=== Model and evaluation on training set ===

Clustered Instances

 0       12 ( 10%)
 1       13 ( 11%)
 2       14 ( 11%)
 3       10 (  8%)
 4        8 (  7%)
 5        8 (  7%)
 6        6 (  5%)
 7        7 (  6%)
 8        6 (  5%)
 9        8 (  7%)
10        9 (  7%)
11        8 (  7%)
12        6 (  5%)
13        7 (  6%)

Unclustered instances : 313

Class attribute: Class
Classes to Clusters:

  0  1  2  3  4  5  6  7  8  9 10 11 12 13  <-- assigned to cluster
  0  0 14  0  0  0  0  0  0  8  9  8  6  7 | democrat
 12 13  0 10  8  8  6  7  6  0  0  0  0  0 | republican

Cluster  0 <-- No class
Cluster  1 <-- republican
Cluster  2 <-- democrat
Cluster  3 <-- No class
Cluster  4 <-- No class
Cluster  5 <-- No class
Cluster  6 <-- No class
Cluster  7 <-- No class
Cluster  8 <-- No class
Cluster  9 <-- No class
Cluster 10 <-- No class
Cluster 11 <-- No class
Cluster 12 <-- No class
Cluster 13 <-- No class

Incorrectly clustered instances :       95.0     21.8391 %
```

➢ **Observations :**

- In this case 13 different clusters are formed and there only 2 class labels available which are republican and democrat.
- The Clusters 1 and 2 will be assigned to these two classes.
- The rest of the clusters are given as no classes.
- The total number of incorrectly clustered instances is 95 which is equal to 21.8391%.

➢ **Comparison Table :**

Algorithms	Iris	Vote
	Incorrectly Clusterd Instances	**Incorrectly Clusterd Instances**
SimpleKMeans	50 , 33.3333%	61, 14.023%
DBSCAN	100, 66.6667%	95, 21.8391%

- ➢ **<u>Analysis :</u>**
- We can notice that for both the algorithms, incorrectly clustered instances reduced from Iris to vote dataset.
- Also we can observe that for KMeans algorithm it reduced by 19% while it reduced by 45% in case of DBSCAN algorithm.
- This is because DBSCAN is designed for more densely populated dataset while KMeans is for simpler datasets.
- Since Vote dataset is densely populated, DBSCAN algorithm performs better for it than Iris dataset.

3. Use 3-4 different sets of parameters for each algorithm (such as number of clusters for kMeans; epsilon and minPoints for DBScan). Experiment until you get much better results than with default. Explain what parameter values gave you best performance and why do you think those values were based based on your understanding of the data and the algorithm.

Solution :

- ➢ **<u>Dataset : Iris</u>**
- ➢ **<u>Using KMeans Algorithm :</u>**

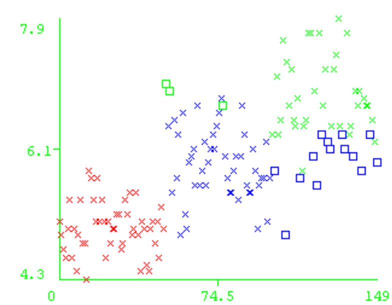
➢ **<u>KMeans Experimentation Results :</u>**

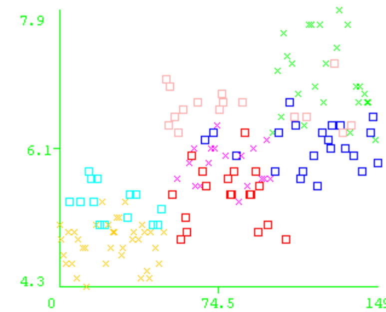
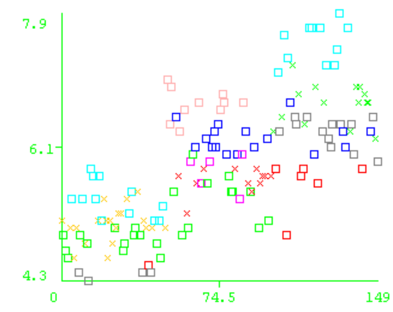
<table>
<tr><th>No of clusters</th><th>Iteration s</th><th>Within cluster (SSE)</th><th>Incorrectly clustered instances</th><th>Cluster details</th><th>Visualize the cluster</th></tr>
<tr><td>3</td><td>6</td><td>6.9981</td><td>17.0,
11.3333 %</td><td>Clustered Instances

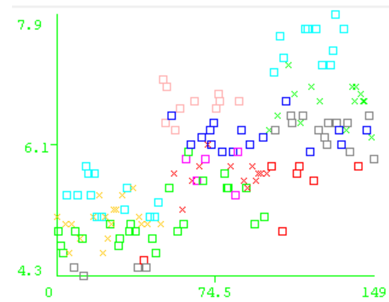
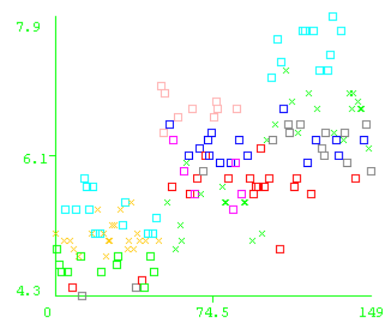
0 61 (41%)
1 50 (33%)
2 39 (26%)

Class attribute: class
Classes to Clusters:

0 1 2 <-- assigned to cluster
0 50 0 | Iris-setosa
47 0 3 | Iris-versicolor
14 0 36 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa
Cluster 2 <-- Iris-virginica</td><td>
</td></tr>
</table>

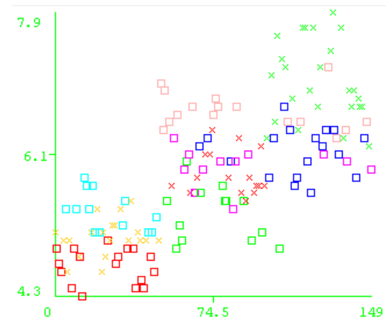
7	7	3.7576	71.0, 47.3333 %	Clustered Instances 0 22 (15%) 1 19 (13%) 2 25 (17%) 3 14 (9%) 4 16 (11%) 5 18 (12%) 6 36 (24%) Class attribute: class Classes to Clusters: 0 1 2 3 4 5 6 <-- assigned to cluster 0 0 0 14 0 0 36 | Iris-setosa 3 18 0 0 11 18 0 | Iris-versicolor 19 1 25 0 5 0 0 | Iris-virginica Cluster 0 <-- No class Cluster 1 <-- No class Cluster 2 <-- Iris-virginica Cluster 3 <-- No class Cluster 4 <-- No class Cluster 5 <-- Iris-versicolor Cluster 6 <-- Iris-setosa	
15	6	2.1605	103.0, 68.6667 %	Clustered Instances 0 9 (6%) 1 12 (8%) 2 15 (10%) 3 12 (8%) 4 10 (7%) 5 5 (3%) 6 20 (13%) 7 1 (1%) 8 11 (7%) 9 4 (3%) 10 8 (5%) 11 6 (4%) 12 13 (9%) 13 11 (7%) 14 13 (9%) Class attribute: class Classes to Clusters: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 <-- assigned to 0 0 0 12 0 0 20 1 0 4 0 0 13 0 0 | Iris-setosa 8 12 0 0 10 5 0 0 11 0 4 0 0 0 0 | Iris-versicolo 1 0 15 0 0 0 0 0 0 0 4 6 0 11 13 | Iris-virginica Cluster 0 <-- No class Cluster 1 <-- Iris-versicolor Cluster 2 <-- Iris-virginica Cluster 3 <-- No class Cluster 4 <-- No class Cluster 5 <-- No class Cluster 6 <-- Iris-setosa Cluster 7 <-- No class Cluster 8 <-- No class Cluster 9 <-- No class Cluster 10 <-- No class Cluster 11 <-- No class Cluster 12 <-- No class Cluster 13 <-- No class Cluster 14 <-- No class	

15	3	2.1605	103.0 68.666 7 %	Clustered Instances 0 8 (5%) 1 13 (9%) 2 15 (10%) 3 13 (9%) 4 10 (7%) 5 5 (3%) 6 19 (13%) 7 1 (1%) 8 11 (7%) 9 4 (3%) 10 8 (5%) 11 6 (4%) 12 13 (9%) 13 11 (7%) 14 13 (9%) Class attribute: class Classes to Clusters: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 <-- assigned to cluste 0 0 0 13 0 0 19 1 0 4 0 0 13 0 0 | Iris-setosa 7 13 0 0 10 5 0 0 11 0 4 0 0 0 0 | Iris-versicolor 1 0 15 0 0 0 0 0 0 0 4 6 0 11 13 | Iris-virginica Cluster 0 <-- No class Cluster 1 <-- Iris-versicolor Cluster 2 <-- Iris-virginica Cluster 3 <-- No class Cluster 4 <-- No class Cluster 5 <-- No class Cluster 6 <-- Iris-setosa Cluster 7 <-- No class Cluster 8 <-- No class Cluster 9 <-- No class Cluster 10 <-- No class Cluster 11 <-- No class Cluster 12 <-- No class Cluster 13 <-- No class Cluster 14 <-- No class	
15	1	3.8797	99.0 66 %	Clustered Instances 0 6 (4%) 1 11 (7%) 2 16 (11%) 3 13 (9%) 4 9 (6%) 5 6 (4%) 6 22 (15%) 7 2 (1%) 8 13 (9%) 9 2 (1%) 10 10 (7%) 11 6 (4%) 12 11 (7%) 13 11 (7%) 14 12 (8%) Class attribute: class Classes to Clusters: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 <-- assigned to cluster 0 0 0 13 0 0 22 2 0 2 0 0 11 0 0 | Iris-setosa 6 11 0 0 9 6 0 0 13 0 4 0 0 0 1 | Iris-versicolor 0 0 16 0 0 0 0 0 0 0 6 6 0 11 11 | Iris-virginica Cluster 0 <-- No class Cluster 1 <-- No class Cluster 2 <-- Iris-virginica Cluster 3 <-- No class Cluster 4 <-- No class Cluster 5 <-- No class Cluster 6 <-- Iris-setosa Cluster 7 <-- No class Cluster 8 <-- Iris-versicolor Cluster 9 <-- No class Cluster 10 <-- No class Cluster 11 <-- No class Cluster 12 <-- No class Cluster 13 <-- No class Cluster 14 <-- No class	

<table>
<tr>
<td>9</td>
<td>7</td>
<td>3.2404</td>
<td>89.0,
59.3333 %</td>
<td>Clustered Instances

0 18 (12%)
1 16 (11%)
2 25 (17%)
3 13 (9%)
4 16 (11%)
5 12 (8%)
6 20 (13%)
7 17 (11%)
8 13 (9%)
Class attribute: class
Classes to Clusters:

0 1 2 3 4 5 6 7 8 <-- assigned to cluster
0 0 0 13 0 0 20 17 0 | Iris-setosa
3 16 0 0 10 9 0 0 12 | Iris-versicolor
15 0 25 0 6 3 0 0 1 | Iris-virginica

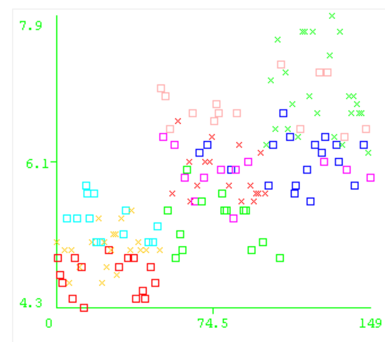
Cluster 0 <-- No class
Cluster 1 <-- Iris-versicolor
Cluster 2 <-- Iris-virginica
Cluster 3 <-- No class
Cluster 4 <-- No class
Cluster 5 <-- No class
Cluster 6 <-- Iris-setosa
Cluster 7 <-- No class
Cluster 8 <-- No class</td>
<td>
</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3.5446</td>
<td>87.0

58%</td>
<td>Clustered Instances

0 18 (12%)
1 17 (11%)
2 25 (17%)
3 13 (9%)
4 14 (9%)
5 12 (8%)
6 21 (14%)
7 16 (11%)
8 14 (9%)

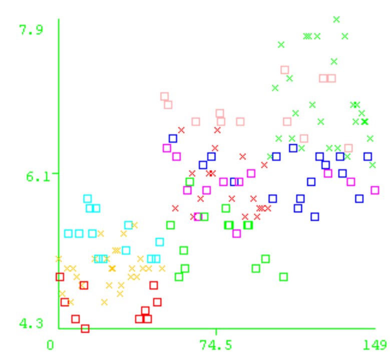
Class attribute: class
Classes to Clusters:

0 1 2 3 4 5 6 7 8 <-- assigned to cluster
0 0 0 13 0 0 21 16 0 | Iris-setosa
3 17 0 0 8 9 0 0 13 | Iris-versicolor
15 0 25 0 6 3 0 0 1 | Iris-virginica

Cluster 0 <-- No class
Cluster 1 <-- Iris-versicolor
Cluster 2 <-- Iris-virginica
Cluster 3 <-- No class
Cluster 4 <-- No class
Cluster 5 <-- No class
Cluster 6 <-- Iris-setosa
Cluster 7 <-- No class
Cluster 8 <-- No class</td>
<td>
</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>5.4606</td>
<td>78.0,
52%</td>
<td>Clustered Instances

0 17 (11%)
1 18 (12%)
2 27 (18%)
3 13 (9%)
4 12 (8%)
5 12 (8%)
6 27 (18%)
7 10 (7%)
8 14 (9%)
Class attribute: class
Classes to Clusters:

0 1 2 3 4 5 6 7 8 <-- assigned to cluster
0 0 0 13 0 0 27 10 0 | Iris-setosa
4 18 0 0 6 9 0 0 13 | Iris-versicolor
13 0 27 0 6 3 0 0 1 | Iris-virginica

Cluster 0 <-- No class
Cluster 1 <-- Iris-versicolor
Cluster 2 <-- Iris-virginica
Cluster 3 <-- No class
Cluster 4 <-- No class
Cluster 5 <-- No class
Cluster 6 <-- Iris-setosa
Cluster 7 <-- No class
Cluster 8 <-- No class</td>
<td>
</td>
</tr>
</table>

- Here we have conducted experiments for different values for parameters number of clusters and iterations.
- We have chosen these because the number of clusters is the basis for whole KMeans algorithm that is the entire KMeans algorithm is dependent on K value that is the number of clusters.

- Also we observed that when the distance function is changed from eucledian to manhattan, there was a increase in the value of SSE. Hence preferred to experiment with eucledian distance itself.

- **<u>Observations :</u>**

- For KMeans it is important to choose the correct value for K or the number of clusters as KMeans is all about predefined number of clusters.
- Above experiments were conducted for different values of number of clusters and number of iterations by keeping the default values for the other parameters.
- We can notice that as the number of clusters increases the value SSE decreases.
- For Iris dataset there has to be minimum of 3 clusters available as there are 3 different classes.
- From the above results we can notice that on single iteration SSE is 5.4606 when K=9 while it is only 3.8797 when K=15.
- Hence we can say that higher number if clusters with optimal number of iterations will fetch us the best results.

- **<u>Comparison with the default value :</u>**
- We have seen that in the experimentation conducted using default values, K=2 and SSE for this is 12.1436 which is high when compared with the number of cluster values in the range 3 to 15 in our experiments.
- Also the number of incorrectly clustered instances for the default value of K=2 is 50 i,e 33.333% whereas the number of incorrectly classified instances decreases for K value of 3 providing better result than the default one.
- As the Iris dataset has 3 classes we get optimal solution when the number of clusters is 3.

  - **<u>Dataset : Iris</u>**

  - **<u>Using DBSCAN Algorithm :</u>**

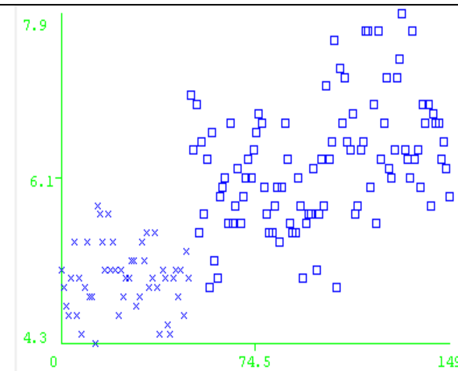
  - **<u>DBCSAN Experimentation Results :</u>**

<table>
<tr><th>Epsilon</th><th>Min Points</th><th>Incorrectly clustered instances</th><th>Cluster formed</th><th>Cluster details</th><th>Visualize the cluster</th></tr>
<tr><td>0.9</td><td>3</td><td>100, 66.6667%</td><td>1</td><td>Clustered Instances

0   150 (100%)

Class attribute: class
Classes to Clusters:

  0 <-- assigned to cluster
 50 | Iris-setosa
 50 | Iris-versicolor
 50 | Iris-virginica

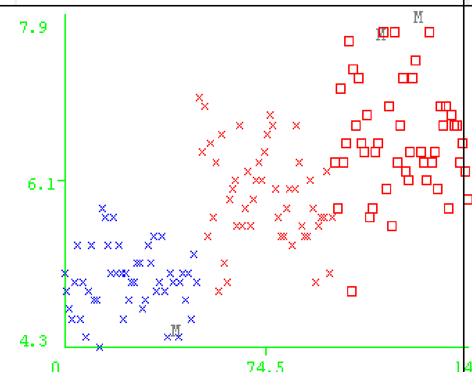
Cluster 0 <-- Iris-setosa</td><td>
</td></tr>
<tr><td>0.2</td><td>5</td><td>48, 32%</td><td>2</td><td>Clustered Instances

0   49 (33%)
1   98 (67%)

Unclustered instances : 3

Class attribute: class
Classes to Clusters:

  0 1 <-- assigned to cluster
 49 0 | Iris-setosa
  0 50 | Iris-versicolor
  0 48 | Iris-virginica

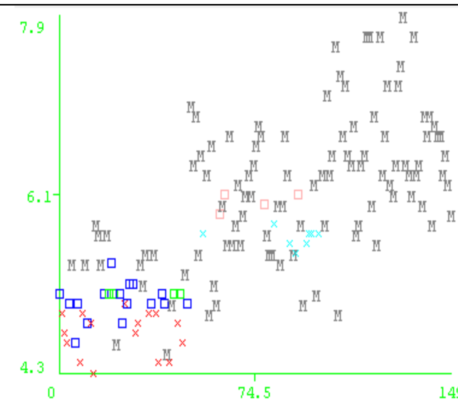
Cluster 0 <-- Iris-setosa
Cluster 1 <-- Iris-versicolor</td><td>
</td></tr>
<tr><td>0.08</td><td>4</td><td>24, 16%</td><td>5</td><td>Clustered Instances

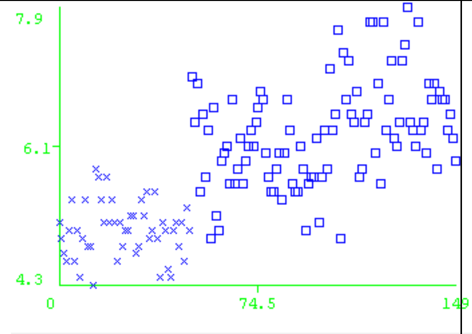
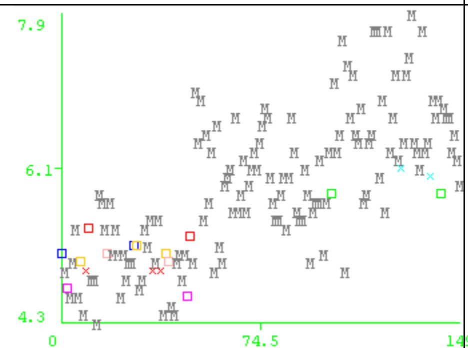
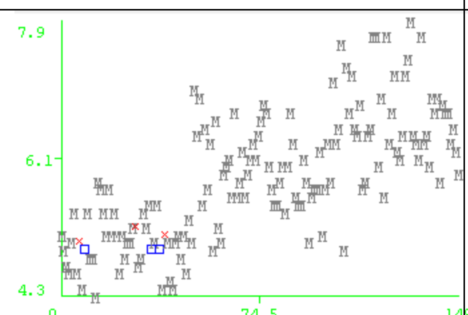
0   16 (33%)
1   16 (33%)
2   4 (8%)
3   8 (17%)
4   4 (8%)

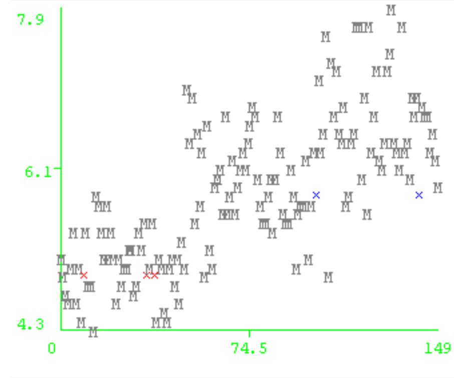
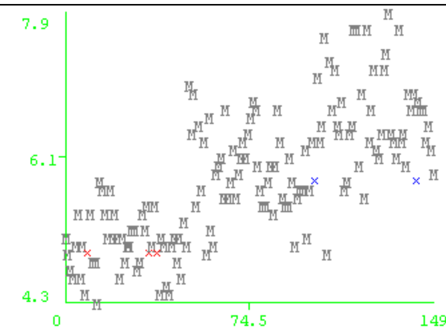
Unclustered instances : 102

Class attribute: class
Classes to Clusters:

  0 1 2 3 4 <-- assigned to cluster
 16 16 4 0 0 | Iris-setosa
  0 0 0 8 4 | Iris-versicolor
  0 0 0 0 0 | Iris-virginica

Cluster 0 <-- No class
Cluster 1 <-- Iris-setosa
Cluster 2 <-- No class
Cluster 3 <-- Iris-versicolor
Cluster 4 <-- No class</td><td>
</td></tr>
</table>

2	1	100, 66.6667%	1	Clustered Instances 0 150 (100%) Class attribute: class Classes to Clusters: 0 <-- assigned to cluster 50 | Iris-setosa 50 | Iris-versicolor 50 | Iris-virginica Cluster 0 <-- Iris-setosa	
0.04	2	13, 8.6667%	8	Clustered Instances 0 2 (11%) 1 2 (11%) 2 2 (11%) 3 2 (11%) 4 2 (11%) 5 2 (11%) 6 3 (17%) 7 3 (17%) Unclustered instances : 132 Class attribute: class Classes to Clusters: 0 1 2 3 4 5 6 7 <-- assigned to cluster 2 2 0 0 2 2 3 3 | Iris-setosa 0 0 0 0 0 0 0 0 | Iris-versicolor 0 0 2 2 0 0 0 0 | Iris-virginica Cluster 0 <-- No class Cluster 1 <-- No class Cluster 2 <-- No class Cluster 3 <-- Iris-virginica Cluster 4 <-- No class Cluster 5 <-- No class Cluster 6 <-- No class Cluster 7 <-- Iris-setosa	
0.04	3	3, 2%	2	Clustered Instances 0 3 (50%) 1 3 (50%) Unclustered instances : 144 Class attribute: class Classes to Clusters: 0 1 <-- assigned to cluster 3 3 | Iris-setosa 0 0 | Iris-versicolor 0 0 | Iris-virginica Cluster 0 <-- No class Cluster 1 <-- Iris-setosa	

0.02	2	0.0, 0%	2	Clustered Instances 0 2 (40%) 1 3 (60%) Unclustered instances : 145 Class attribute: class Classes to Clusters: 0 1 <-- assigned to cluster 0 3 | Iris-setosa 0 0 | Iris-versicolor 2 0 | Iris-virginica Cluster 0 <-- Iris-virginica Cluster 1 <-- Iris-setosa	
0.01	2	0.0, 0%	2	Clustered Instances 0 2 (40%) 1 3 (60%) Unclustered instances : 145 Class attribute: class Classes to Clusters: 0 1 <-- assigned to cluster 0 3 | Iris-setosa 0 0 | Iris-versicolor 2 0 | Iris-virginica Cluster 0 <-- Iris-virginica Cluster 1 <-- Iris-setosa	

- **<u>Observations :</u>**

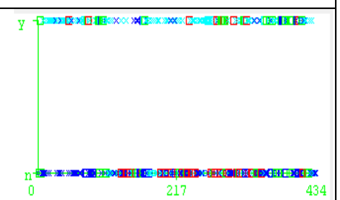
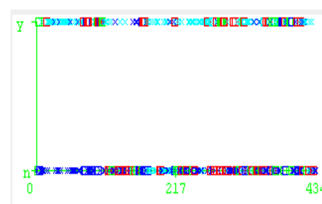
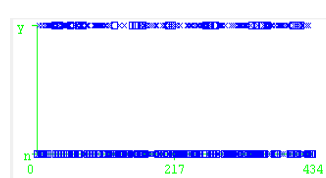
- Here we have conducted experiments for different values of parameters of epsilon and minpoints.
- From the above experiment conducted for various values of epsilon and minpoints, we can say that as the epsilon value decreases the number of incorrectly classified instances also decreases.
- From the above results we can say that when epsilon value is 0.02 and 0.01 and the minpoints value is 2, we get the best solution where there are no incorrectly classified instances.
- For low value of epsilon if the value of minpoints is very less ,then the number of incorrectly classified instances increases.
- We notice that for epsilon value of 0.02 and 0.01, and the value of minpoints is 1 then the number of incorrectly classified instances increased to 144.
- For a given epsilon value as the minpoints increases incorrectly classified instances decreases.
- From the above tabulated results for the epsilon value of 0.04 the number of incorrectly classified instances is 13 when minpoint is 2 while it is 3 when minpoint is 3.
- The above case hold good only for smaller epsilon values.

- ➢ **<u>Comparison with the default value of the parameters :</u>**
- For the default values of epsilon and minpoints i,e 0.9 and 6, the number of incorrectly classified instances is 100 that is 66.6667%.
- In our experimentation, we have got better results when experimenting with lower values of epsilon which in turn decreases the number of incorrectly classified instances.

➢ **<u>Dataset : Vote</u>**

➢ **<u>Using KMeans Algorithm :</u>**

➢ **<u>KMeans Experimentation Results :</u>**

No of clusters	**Iterations**	**dontReplaceMissingValues**	**Within cluster(SSE)**	**InCorrectly clustered instances**	**Picture of cluster details**	**Visualize the cluster**
4	3	False	1225.0	142.0 32.6437 %	Clustered Instances 0 167 (38%) 1 52 (12%) 2 61 (14%) 3 155 (36%) Class attribute: Class Classes to Clusters: 0 1 2 3 <-- assigned to cluster 22 45 52 148 | democrat 145 7 9 7 | republican Cluster 0 <-- republican Cluster 1 <-- No class Cluster 2 <-- No class Cluster 3 <-- democrat	
4	3	true	1517.0	145.0 33.3333 %	Clustered Instances 0 185 (43%) 1 76 (17%) 2 36 (8%) 3 138 (32%) Class attribute: Class Classes to Clusters: 0 1 2 3 <-- assigned to clust 31 64 36 136 | democrat 154 12 0 2 | republican Cluster 0 <-- republican Cluster 1 <-- No class Cluster 2 <-- No class Cluster 3 <-- democrat	
1	1	false	3173.0	168.0 38.6207 %	Clustered Instances 0 435 (100%) Class attribute: Class Classes to Clusters: 0 <-- assigned to cluster 267 | democrat 168 | republican Cluster 0 <-- democrat	

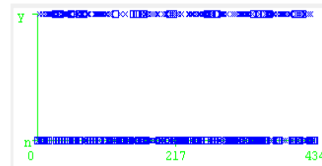
<table>
<tr><td>1</td><td>1</td><td>true</td><td>3711.0</td><td>168.0

38.620
7 %</td><td>Clustered Instances

0 435 (100%)

Class attribute: Class
Classes to Clusters:

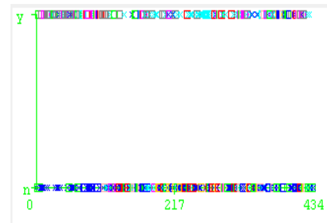
0 <-- assigned to cluster
267 | democrat
168 | republican

Cluster 0 <-- democrat</td><td></td></tr>
<tr><td>10</td><td>3</td><td>false</td><td>971.0</td><td>227.0
52.1839 %</td><td>Clustered Instances

0 146 (34%)
1 28 (6%)
2 18 (4%)
3 90 (21%)
4 33 (8%)
5 27 (6%)
6 18 (4%)
7 7 (2%)
8 27 (6%)
9 41 (9%)

Class attribute: Class
Classes to Clusters:

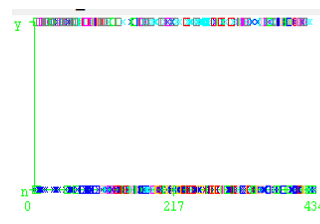
0 1 2 3 4 5 6 7 8 9 <-- assign
22 22 18 84 28 26 18 6 2 41 | democrat
124 6 0 6 5 1 0 1 25 0 | republica

Cluster 0 <-- republican
Cluster 1 <-- No class
Cluster 2 <-- No class
Cluster 3 <-- democrat
Cluster 4 <-- No class
Cluster 5 <-- No class
Cluster 6 <-- No class
Cluster 7 <-- No class
Cluster 8 <-- No class
Cluster 9 <-- No class</td><td></td></tr>
<tr><td>10</td><td>3</td><td>true</td><td>1294.0</td><td>231.0
53.1034 %</td><td>Clustered Instances

0 104 (24%)
1 18 (4%)
2 28 (6%)
3 126 (29%)
4 21 (5%)
5 15 (3%)
6 27 (6%)
7 3 (1%)
8 76 (17%)
9 17 (4%)

Class attribute: Class
Classes to Clusters:

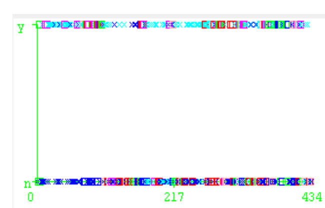
0 1 2 3 4 5 6 7 8 9 <-- assigne
23 6 27 123 20 15 26 3 7 17 | democrat
81 12 1 3 1 0 1 0 69 0 | republican

Cluster 0 <-- republican
Cluster 1 <-- No class
Cluster 2 <-- No class
Cluster 3 <-- democrat
Cluster 4 <-- No class
Cluster 5 <-- No class
Cluster 6 <-- No class
Cluster 7 <-- No class
Cluster 8 <-- No class
Cluster 9 <-- No class</td><td></td></tr>
<tr><td>6</td><td>5</td><td>False</td><td>1098.0</td><td>173.0
39.7701 %</td><td>Clustered Instances

0 168 (39%)
1 47 (11%)
2 37 (9%)
3 122 (28%)
4 33 (8%)
5 28 (6%)

Class attribute: Class
Classes to Clusters:

0 1 2 3 4 5 <-- assigned to
26 38 35 120 21 27 | democrat
142 9 2 2 12 1 | republican

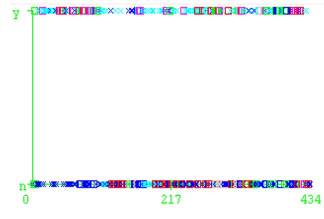
Cluster 0 <-- republican
Cluster 1 <-- No class
Cluster 2 <-- democrat
Cluster 3 <-- democrat
Cluster 4 <-- No class
Cluster 5 <-- No class</td><td></td></tr>
</table>

<table>
<tr><td>6</td><td>5</td><td>true</td><td>1482.0</td><td>169.0
38.8506 %</td><td>Clustered Instances

0 184 (42%)
1 62 (14%)
2 35 (8%)
3 115 (26%)
4 18 (4%)
5 21 (5%)

Class attribute: Class
Classes to Clusters:

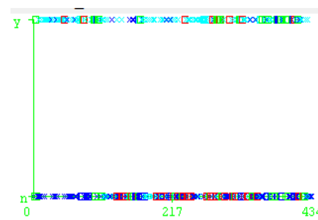
0 1 2 3 4 5 <-- assigned to cl
30 54 34 112 18 19 | democrat
154 8 1 3 0 2 | republican

Cluster 0 <-- republican
Cluster 1 <-- No class
Cluster 2 <-- No class
Cluster 3 <-- democrat
Cluster 4 <-- No class
Cluster 5 <-- No class</td><td></td></tr>
<tr><td>4</td><td>2</td><td>false</td><td>1225.0</td><td>142.0
32.6437 %</td><td>Clustered Instances

0 167 (38%)
1 52 (12%)
2 61 (14%)
3 155 (36%)

Class attribute: Class
Classes to Clusters:

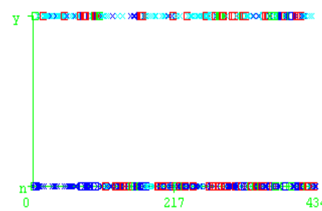
0 1 2 3 <-- assigned to clu
22 45 52 148 | democrat
145 7 9 7 | republican

Cluster 0 <-- republican
Cluster 1 <-- No class
Cluster 2 <-- No class
Cluster 3 <-- democrat</td><td></td></tr>
<tr><td>4</td><td>2</td><td>True</td><td>1525.0</td><td>151.0
34.7126 %</td><td>Clustered Instances

0 187 (43%)
1 84 (19%)
2 31 (7%)
3 133 (31%)

Class attribute: Class
Classes to Clusters:

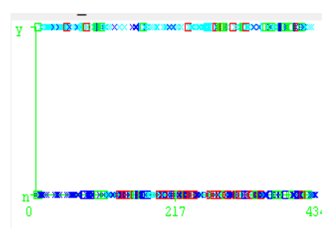
0 1 2 3 <-- assigned to clu
34 72 30 131 | democrat
153 12 1 2 | republican

Cluster 0 <-- republican
Cluster 1 <-- No class
Cluster 2 <-- No class
Cluster 3 <-- democrat</td><td></td></tr>
<tr><td>4</td><td>1</td><td>false</td><td>1839.0</td><td>142.0
32.6437 %</td><td>Clustered Instances

0 167 (38%)
1 52 (12%)
2 61 (14%)
3 155 (36%)

Class attribute: Class
Classes to Clusters:

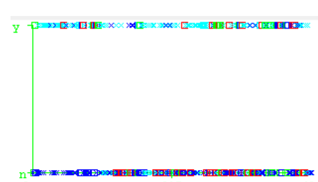
0 1 2 3 <-- assigned to cluster
22 45 52 148 | democrat
145 7 9 7 | republican

Cluster 0 <-- republican
Cluster 1 <-- No class
Cluster 2 <-- No class
Cluster 3 <-- democrat</td><td></td></tr>
<tr><td>4</td><td>1</td><td>true</td><td>2226.0</td><td>137.0
31.4943 %</td><td>Clustered Instances

0 185 (43%)
1 60 (14%)
2 38 (9%)
3 152 (35%)

Class attribute: Class
Classes to Clusters:

0 1 2 3 <-- assigned to cluster
32 54 36 145 | democrat
153 6 2 7 | republican

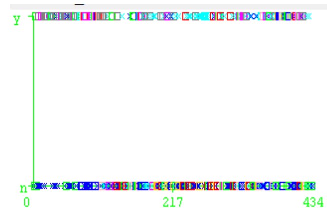
Cluster 0 <-- republican
Cluster 1 <-- No class
Cluster 2 <-- No class
Cluster 3 <-- democrat</td><td></td></tr>
</table>

<table>
<tr><td>10</td><td>2</td><td>false</td><td>971</td><td>227.0
52.1839 %</td><td>Clustered Instances

0 146 (34%)
1 28 (6%)
2 18 (4%)
3 90 (21%)
4 33 (8%)
5 27 (6%)
6 18 (4%)
7 7 (2%)
8 27 (6%)
9 41 (9%)

Class attribute: Class
Classes to Clusters:

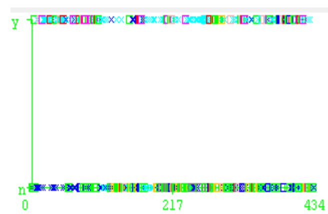
0 1 2 3 4 5 6 7 8 9 <-- as
22 22 18 84 28 26 18 6 2 41 | democ
124 6 0 6 5 1 0 1 25 0 | repub

Cluster 0 <-- republican
Cluster 1 <-- No class
Cluster 2 <-- No class
Cluster 3 <-- democrat
Cluster 4 <-- No class
Cluster 5 <-- No class
Cluster 6 <-- No class
Cluster 7 <-- No class
Cluster 8 <-- No class
Cluster 9 <-- No class</td><td></td></tr>
<tr><td>10</td><td>2</td><td>true</td><td>1345.0</td><td>230.0
52.8736 %</td><td>Clustered Instances

0 103 (24%)
1 16 (4%)
2 28 (6%)
3 128 (29%)
4 21 (5%)
5 16 (4%)
6 28 (6%)
7 2 (0%)
8 76 (17%)
9 17 (4%)

Class attribute: Class
Classes to Clusters:

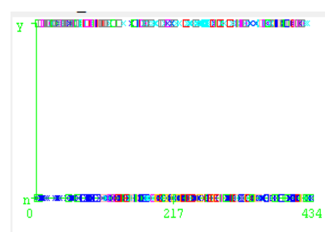
0 1 2 3 4 5 6 7 8 9 <-- ass
23 4 27 125 20 16 26 2 7 17 | democr
80 12 1 3 1 0 2 0 69 0 | republ

Cluster 0 <-- republican
Cluster 1 <-- No class
Cluster 2 <-- No class
Cluster 3 <-- democrat
Cluster 4 <-- No class
Cluster 5 <-- No class
Cluster 6 <-- No class
Cluster 7 <-- No class
Cluster 8 <-- No class
Cluster 9 <-- No class</td><td></td></tr>
<tr><td>10</td><td>1</td><td>false</td><td>1585.0</td><td>227.0
52.1839 %</td><td>Clustered Instances

0 146 (34%)
1 28 (6%)
2 18 (4%)
3 90 (21%)
4 33 (8%)
5 27 (6%)
6 18 (4%)
7 7 (2%)
8 27 (6%)
9 41 (9%)

Class attribute: Class
Classes to Clusters:

0 1 2 3 4 5 6 7 8 9 <-- assign
22 22 18 84 28 26 18 6 2 41 | democrat
124 6 0 6 5 1 0 1 25 0 | republica

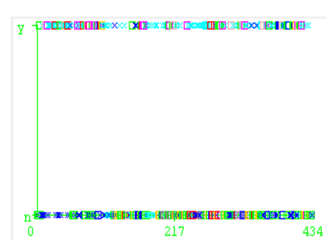
Cluster 0 <-- republican
Cluster 1 <-- No class
Cluster 2 <-- No class
Cluster 3 <-- democrat
Cluster 4 <-- No class
Cluster 5 <-- No class
Cluster 6 <-- No class
Cluster 7 <-- No class
Cluster 8 <-- No class
Cluster 9 <-- No class</td><td></td></tr>
<tr><td>10</td><td>1</td><td>true</td><td>2013.0</td><td>214.0

49.195
4 %</td><td>Clustered Instances

0 104 (24%)
1 18 (4%)
2 28 (6%)
3 126 (29%)
4 21 (5%)
5 15 (3%)
6 27 (6%)
7 3 (1%)
8 76 (17%)
9 17 (4%)

Class attribute: Class
Classes to Clusters:

0 1 2 3 4 5 6 7 8 9 <-- assig
23 6 27 123 20 15 26 3 7 17 | democrat
81 12 1 3 1 0 1 0 69 0 | republic

Cluster 0 <-- republican
Cluster 1 <-- No class
Cluster 2 <-- No class
Cluster 3 <-- democrat
Cluster 4 <-- No class
Cluster 5 <-- No class
Cluster 6 <-- No class
Cluster 7 <-- No class
Cluster 8 <-- No class
Cluster 9 <-- No class</td><td></td></tr>
</table>

- Here we have conducted experiments for different values for parameters number of clusters iterations and dontReplace Missing Values.
- We have chosen these because the number of clusters is the basis for whole KMeans algorithm that is the entire KMeans algorithm is dependent on K value that is the number of clusters.

- **<u>Observations :</u>**

- From the above results we can say that as the number of clusters increases the value of SSE decreases.
- For smaller number of clusters higher the value of iterations, lower is the SSE.
- We can notice that in our experiment results when the number of clusters is 4 and the number of iterations is 1, the value of SSE is 1839.0
- For the same number of iterations K=10, SSE is 1584.0.
- When K=4, and the number of iterations is 2 the value of SSE is 1225.0 But it is 1839.0 when number of iteration is 1.
- When missing values are replaced with mean we are able to get better results.
- For instance, when K=4 and number of iterations is 1, SSE is 1839.0 when dontReplaceMissingValues is set to false. But it increases to 2226.0 when donReplaceMissingValues is set to true.

- **<u>Comparison With default values :</u>**
- For the default values, the value of SSE was 1449.0.
- Default values for K was 2 and maxiteration was 500 and dontreplaceMissingValues was false.
- In our experimentation results it can be seen that as the number of clusters increase, the value of SSE decreases.
- It was also noticed when maxIteration is very high variation in maxIteration dosent effect SSE.
- For default values the number of incorrectly classified instances were 16 and in our experimentation it has increased to 142 at K=4 and 168 at K=1.
- We can also notice that the default value of ‘false’ for dontReplaceMissingValues is better.

- **Dataset : Vote**
- **Using DBSCAN Algorithm :**
- **DBSCAN Experimentation Results :**

Epsilon	Min Points	Incorrectly Clusters Instances	Clusters Formed	Unclustered Instances	Cluster Details
0.09	4	122, 28.046%	20	286	Cluster 0 <-- No class Cluster 1 <-- republican Cluster 2 <-- democrat Cluster 3 <-- No class Cluster 4 <-- No class Cluster 5 <-- No class Cluster 6 <-- No class Cluster 7 <-- No class Cluster 8 <-- No class Cluster 9 <-- No class Cluster 10 <-- No class Cluster 11 <-- No class Cluster 12 <-- No class Cluster 13 <-- No class Cluster 14 <-- No class Cluster 15 <-- No class Cluster 16 <-- No class Cluster 17 <-- No class Cluster 18 <-- No class Cluster 19 <-- No class
0.5	4	122, 28.046%	20	286	Cluster 0 <-- No class Cluster 1 <-- republican Cluster 2 <-- democrat Cluster 3 <-- No class Cluster 4 <-- No class Cluster 5 <-- No class Cluster 6 <-- No class Cluster 7 <-- No class Cluster 8 <-- No class Cluster 9 <-- No class Cluster 10 <-- No class Cluster 11 <-- No class Cluster 12 <-- No class Cluster 13 <-- No class Cluster 14 <-- No class Cluster 15 <-- No class Cluster 16 <-- No class Cluster 17 <-- No class Cluster 18 <-- No class Cluster 19 <-- No class

<table>
<tr><td>0.05</td><td>2</td><td>171,
39.3103%</td><td>42</td><td>237</td><td>Cluster 0 <-- No class
Cluster 1 <-- republican
Cluster 2 <-- No class
Cluster 3 <-- democrat
Cluster 4 <-- No class
Cluster 5 <-- No class
Cluster 6 <-- No class
Cluster 7 <-- No class
Cluster 8 <-- No class
Cluster 9 <-- No class
Cluster 10 <-- No class
Cluster 11 <-- No class
Cluster 12 <-- No class
Cluster 13 <-- No class
Cluster 14 <-- No class
Cluster 15 <-- No class
Cluster 16 <-- No class
Cluster 17 <-- No class
Cluster 18 <-- No class
Cluster 19 <-- No class
Cluster 20 <-- No class
Cluster 21 <-- No class
Cluster 22 <-- No class
Cluster 23 <-- No class
Cluster 24 <-- No class
Cluster 25 <-- No class
Cluster 26 <-- No class
Cluster 27 <-- No class
Cluster 28 <-- No class
Cluster 29 <-- No class
Cluster 30 <-- No class
Cluster 31 <-- No class
Cluster 32 <-- No class
Cluster 33 <-- No class
Cluster 34 <-- No class
Cluster 35 <-- No class
Cluster 36 <-- No class
Cluster 37 <-- No class
Cluster 38 <-- No class
Cluster 39 <-- No class
Cluster 40 <-- No class
Cluster 41 <-- No class</td></tr>
<tr><td>0.9</td><td>10</td><td>22,
5.0575%</td><td>4</td><td>386</td><td>Clustered Instances

0   12 (24%)
1   13 (27%)
2   14 (29%)
3   10 (20%)

Unclustered instances : 386

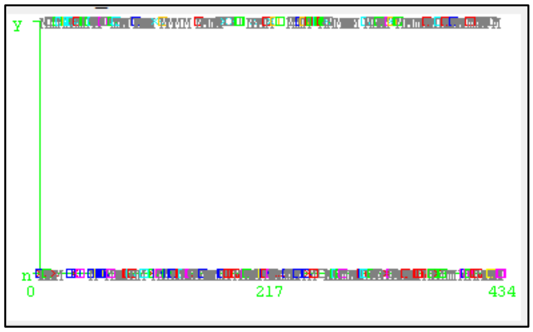
Class attribute: Class
Classes to Clusters:

  0 1 2 3 <-- assigned to cluster
  0 0 14 0 | democrat
 12 13 0 10 | republican

Cluster 0 <-- No class
Cluster 1 <-- republican
Cluster 2 <-- democrat
Cluster 3 <-- No class</td></tr>
<tr><td>0.5</td><td>6</td><td>95,
21.8391%</td><td>14</td><td>313</td><td>Cluster 0 <-- No class
Cluster 1 <-- republican
Cluster 2 <-- democrat
Cluster 3 <-- No class
Cluster 4 <-- No class
Cluster 5 <-- No class
Cluster 6 <-- No class
Cluster 7 <-- No class
Cluster 8 <-- No class
Cluster 9 <-- No class
Cluster 10 <-- No class
Cluster 11 <-- No class
Cluster 12 <-- No class
Cluster 13 <-- No class</td></tr>
</table>

0.05	4	122, 28.046%	20	286	Cluster 0 <-- No class Cluster 1 <-- republican Cluster 2 <-- democrat Cluster 3 <-- No class Cluster 4 <-- No class Cluster 5 <-- No class Cluster 6 <-- No class Cluster 7 <-- No class Cluster 8 <-- No class Cluster 9 <-- No class Cluster 10 <-- No class Cluster 11 <-- No class Cluster 12 <-- No class Cluster 13 <-- No class Cluster 14 <-- No class Cluster 15 <-- No class Cluster 16 <-- No class Cluster 17 <-- No class Cluster 18 <-- No class Cluster 19 <-- No class
1	6	95, 21.8391%	14	313	Cluster 0 <-- No class Cluster 1 <-- republican Cluster 2 <-- democrat Cluster 3 <-- No class Cluster 4 <-- No class Cluster 5 <-- No class Cluster 6 <-- No class Cluster 7 <-- No class Cluster 8 <-- No class Cluster 9 <-- No class Cluster 10 <-- No class Cluster 11 <-- No class Cluster 12 <-- No class Cluster 13 <-- No class
0.09	2	171, 39.3103%	42	237	Cluster 0 <-- No class Cluster 1 <-- republican Cluster 2 <-- No class Cluster 3 <-- democrat Cluster 4 <-- No class Cluster 5 <-- No class Cluster 6 <-- No class Cluster 7 <-- No class Cluster 8 <-- No class Cluster 9 <-- No class Cluster 10 <-- No class Cluster 11 <-- No class Cluster 12 <-- No class Cluster 13 <-- No class Cluster 14 <-- No class Cluster 15 <-- No class Cluster 16 <-- No class Cluster 17 <-- No class Cluster 18 <-- No class Cluster 19 <-- No class Cluster 20 <-- No class Cluster 21 <-- No class Cluster 22 <-- No class Cluster 23 <-- No class Cluster 24 <-- No class Cluster 25 <-- No class Cluster 26 <-- No class Cluster 27 <-- No class Cluster 28 <-- No class Cluster 29 <-- No class Cluster 30 <-- No class Cluster 31 <-- No class Cluster 32 <-- No class Cluster 33 <-- No class Cluster 34 <-- No class Cluster 35 <-- No class Cluster 36 <-- No class Cluster 37 <-- No class Cluster 38 <-- No class Cluster 39 <-- No class Cluster 40 <-- No class Cluster 41 <-- No class

➢ **Using the Visualize tab we observe the following fig. :**



➢ **Observations :**

- Here we conducted experiments for different values of parameters of epsilon and minpoints.
- We can notice that as the epsilon value decreases the number of incorrectly classified instances also decreases.
- From the above tabulated results we notice that for the epsilon value of 0.09 the number of incorrectly classified instances increases from 122 at minpoint 4 to 171 at minpoint 2.
- In this case we did not find any different results on keeping the minpoints constant and varying the epsilon.

➢ **Comparison with the default values :**

➢ When experimented with the default value of epsilon and minpoints, the number of incorrectly classified instances is 95 which is equal to 21.8391%.

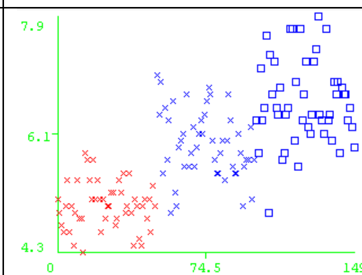
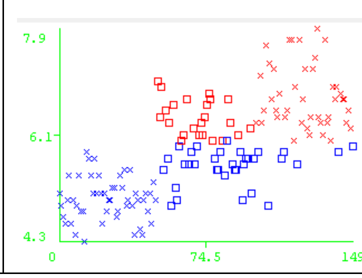
➢ When we kept the value of minpoint as in default and changed the epsilon we didn't find any differentiating results.

➢ But on keeping epsilon as in default and increasing the minpoints did decrease the number of incorrectly classified instances from 95 to 22.

4. **Select some attributes based on your analysis in 1) and use only them during clustering. Analyze the results. How are the results different from 2)? Is it what you expected based on your analysis of the data?**

Solution :

- **Dataset : Iris**
- **Using SimpleKMeans Algorithm :**

Algorithm	No. of Iterations	Within Cluster SSE	Incorrectly clustered instances	Attribute Information	Cluster details	Visualize the cluster
KMeans	4	6.3369	50, 33.333%	Relation: iris Instances: 150 Attributes: 5 sepalwidth petalwidth Ignored: sepallength petallength class	Cluster centroids: Cluster# Attribute Full Data 0 1 (150) (100) (50) == sepalwidth 3.054 2.872 3.418 petalwidth 1.1987 1.676 0.244 === Model and evaluation on training set === Clustered Instances 0 100 (67%) 1 50 (33%) Class attribute: class Classes to Clusters: 0 1 <-- assigned to cluster 0 50 | Iris-setosa 50 0 | Iris-versicolor 50 0 | Iris-virginica Cluster 0 <-- Iris-versicolor Cluster 1 <-- Iris-setosa	
KMeans	4	7.0810	57, 38%	Relation: iris Instances: 150 Attributes: 5 sepallength sepalwidth Ignored: petallength petalwidth class	Cluster centroids: Cluster# Attribute Full Data 0 1 (150) (84) (66) == sepallength 5.8433 5.2333 6.6197 sepalwidth 3.054 3.1286 2.9591	

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td>Clustered Instances

0 84 (56%)
1 66 (44%)

Class attribute: class
Classes to Clusters:

0 1 <-- assigned to cluster
50 0 | Iris-setosa
27 23 | Iris-versicolor
7 43 | Iris-virginica

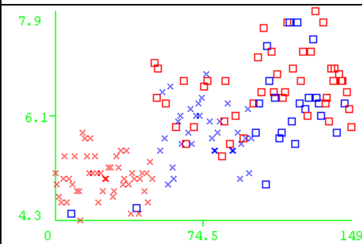
Cluster 0 <-- Iris-setosa
Cluster 1 <-- Iris-virginica</td><td></td></tr>
<tr><td>KMeans</td><td>2</td><td>2.1322</td><td>68,
45.3333%</td><td>Relation: iris
Instances: 150
Attributes: 5
sepalwidth
Ignored:
sepallength
petallength
petalwidth
class</td><td>Cluster centroids:
Cluster#
Attribute Full Data 0 1
(150) (57) (93)
==
sepalwidth 3.054 2.6404 3.3075

Clustered Instances

0 57 (38%)
1 93 (62%)

Class attribute: class
Classes to Clusters:

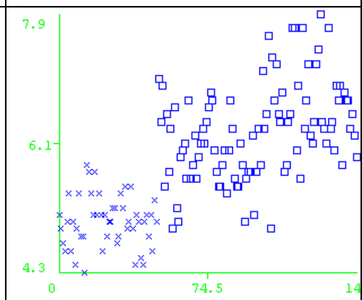
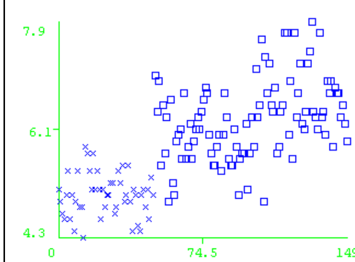
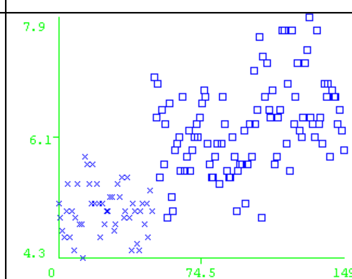
0 1 <-- assigned to cluster
2 48 | Iris-setosa
34 16 | Iris-versicolor
21 29 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa</td><td>
</td></tr>
</table>

- **Observations :**

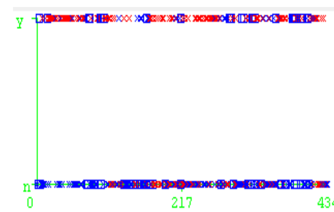
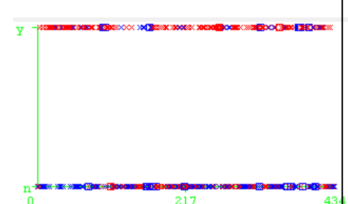
- In the first case ,have considered the attributes of sepal width and petal width for our experimentation and ignored the rest.
- Incase of SimpleKMeans the value of SSE decreased to 6.3369 from 12.1436 when all the attributes were considered and the number of incorrectly clustered instances remains the same as in the default one.
- In the second case have considered the attributes of sepal width and sepal width for our experimentation and ignored the rest.
- The value of SSE decreased to 7.0810 from 12.1436 when all the attributes were considered and the number of incorrectly clustered instances increased to 57, 38% when compared with the default one.
- In the third case, have considered the attribute of sepal width for our experimentation and ignored the rest.
- The value of SSE decreased to 2.1322 from 12.1436 when all the attributes were considered and the number of incorrectly clustered instances increased to 68, 45.33% when compared with the default one.

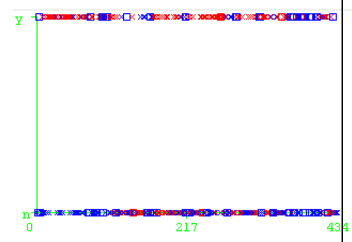
- **Dataset : Iris**
- **Using DBSCAN Algorithm :**

Algorithm	No. of Clusters	Epsilon / Minpoints	Incorrectly clustered instances	Attribute Information	Cluster details	Visualize the cluster
DBSCAN	1	0.9 / 6	100, 66.6667%	Relation: iris Instances: 150 Attributes: 5 sepalwidth petalwidth Ignored: sepallength petallength class	Clustered Instances 0 150 (100%) Class attribute: class Classes to Clusters: 0 <-- assigned to cluster 50 | Iris-setosa 50 | Iris-versicolor 50 | Iris-virginica Cluster 0 <-- Iris-setosa	
DBSCAN	1	0.9 / 6	100, 66.6667%	Relation: iris Instances: 150 Attributes: 5 sepallength sepalwidth Ignored: petallength petalwidth class	Clustered Instances 0 150 (100%) Class attribute: class Classes to Clusters: 0 <-- assigned to cluster 50 | Iris-setosa 50 | Iris-versicolor 50 | Iris-virginica Cluster 0 <-- Iris-setosa	
DBSCAN	1	0.9 / 6	100, 66.6667%	Relation: iris Instances: 150 Attributes: 5 sepalwidth Ignored: sepallength petallength petalwidth class	Clustered Instances 0 150 (100%) Class attribute: class Classes to Clusters: 0 <-- assigned to cluster 50 | Iris-setosa 50 | Iris-versicolor 50 | Iris-virginica Cluster 0 <-- Iris-setosa	

- **Observations :**
- In case of DBSCAN there was no change in the number of incorrectly clustered instances when compared with the default one even when different attributes where considered as shown in the above experimentation results.

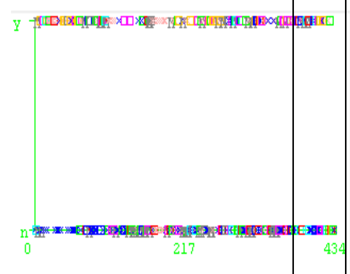
- **Dataset : Vote**
- **Using SimpleKMeans Algorithm :**

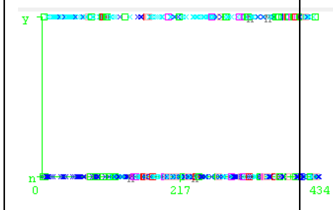
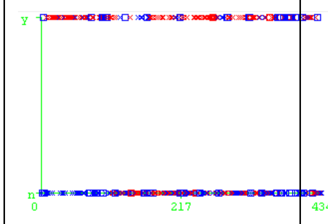
No. of Iterations	Within Cluster SSE	Incorrectly clustered instances	Attribute Information	Cluster details	Visualize the cluster
3	427	78, 17.931	Relation: vote Instances: 435 Attributes: 17 adoption-of-the-budget-resolution physician-fee-freeze religious-groups-in-schools anti-satellite-test-ban crime duty-free-exports Ignored: handicapped-infants water-project-cost-sharing el-salvador-aid aid-to-nicaraguan-contras mx-missile immigration synfuels-corporation-cutback education-spending superfund-right-to-sue export-administration-act-south-africa Class	Cluster centroids: Cluster# Attribute Full Data 0 1 (435) (244) (191) == adoption-of-the-budget-resolution y n y physician-fee-freeze n y n religious-groups-in-schools y y n anti-satellite-test-ban y n y crime y y n duty-free-exports n n y Clustered Instances 0 244 (56%) 1 191 (44%) Class attribute: Class Classes to Clusters: 0 1 <-- assigned to cluster 77 190 | democrat 167 1 | republican Cluster 0 <-- republican Cluster 1 <-- democrat	
3	105.0	34, 7.8161	Relation: vote Instances: 435 Attributes: 17 adoption-of-the-budget-resolution physician-fee-freeze el-salvador-aid Ignored: handicapped-infants water-project-cost-sharing religious-groups-in-schools anti-satellite-test-ban aid-to-nicaraguan-contras mx-missile immigration synfuels-corporation-cutback education-spending superfund-right-to-sue crime duty-free-exports export-administration-act-south-africa Class	Cluster centroids: Cluster# Attribute Full Data 0 1 (435) (186) (249) == adoption-of-the-budget-resolution y n y physician-fee-freeze n y n el-salvador-aid y y n Clustered Instances 0 186 (43%) 1 249 (57%) Class attribute: Class Classes to Clusters: 0 1 <-- assigned to cluster 26 241 | democrat 160 8 | republican Cluster 0 <-- republican Cluster 1 <-- democrat	

2	0.0	75, 17.2414 %	Relation: vote Instances: 435 Attributes: 17 el-salvador-aid Ignored: handicapped-infants water-project-cost-sharing adoption-of-the-budget-resolution physician-fee-freeze religious-groups-in-schools anti-satellite-test-ban aid-to-nicaraguan-contras mx-missile immigration synfuels-corporation-cutback education-spending superfund-right-to-sue crime duty-free-exports export-administration-act-south-africa Class	Clustered Instances 0 227 (52%) 1 208 (48%) Class attribute: Class Classes to Clusters: 0 1 <-- assigned to cluster 67 200 | democrat 160 8 | republican Cluster 0 <-- republican Cluster 1 <-- democrat	

- **Observations :**
- In the first case, attributes adoption of the budget resolution, religious groups in school, anti satellite test ban, crime, physician fee freeze and duty free exports were considered for the experimentation and the rest were ignored.
- Incase of simpleKMeans algorithm the value of SSE decreased from 1449 in the default to 427 while the number of incorrectly clustered instances increased from 61 to 78.
- In the second case, attributes adoption of the budget resolution, physician fee freeze and el-salvador-aid were considered for the experimentation and the rest were ignored.
- The value of SSE decreased from 1449 in the default to 105 while the number of incorrectly clustered instances decreased from 61 to 34.
- In the third case only the attribute el-salvador-aid was considered.
- Here the value of SSE is 0 and the number of incorrectly clustered instances inceased from 61 to 75.
- We can say that el-salvador-aid is one of the important attribute to look into for in Vote dataset.

- **Dataset : Vote**
- **Using DBSCAN Algorithm :**

No. of Clusters	Epsilon / Minpoints	Incorrectly clustered instances	Attribute Information	Cluster details	Visualize the cluster
16	0.9 / 6	185, 42.5287 %	Relation: vote Instances: 435 Attributes: 17 adoption-of-the-budget-resolution physician-fee-freeze religious-groups-in-schools anti-satellite-test-ban crime duty-free-exports Ignored: handicapped-infants water-project-cost-sharing el-salvador-aid aid-to-nicaraguan-contras mx-missile immigration synfuels-corporation-cutback education-spending superfund-right-to-sue export-administration-act-south-africa Class	Clustered Instances 0 110 (29%) 1 6 (2%) 2 22 (6%) 3 14 (4%) 4 82 (22%) 5 25 (7%) 6 27 (7%) 7 10 (3%) 8 8 (2%) 9 6 (2%) 10 15 (4%) 11 6 (2%) 12 15 (4%) 13 8 (2%) 14 9 (2%) 15 11 (3%) Unclustered instances : 61 Class attribute: Class Classes to Clusters: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 <-- assigned to cluster 3 6 20 14 82 25 27 6 8 6 15 1 0 7 0 11 | democrat 107 0 2 0 0 0 0 4 0 0 0 5 15 1 9 0 | republican Cluster 0 <-- republican Cluster 1 <-- No class Cluster 2 <-- No class Cluster 3 <-- No class Cluster 4 <-- democrat Cluster 5 <-- No class Cluster 6 <-- No class Cluster 7 <-- No class Cluster 8 <-- No class Cluster 9 <-- No class Cluster 10 <-- No class Cluster 11 <-- No class Cluster 12 <-- No class Cluster 13 <-- No class Cluster 14 <-- No class Cluster 15 <-- No class	

6	0.9 / 6	102, 23.4483 %	Relation: vote Instances: 435 Attributes: 17 adoption-of-the-budget-resolution physician-fee-freeze el-salvador-aid Ignored: handicapped-infants water-project-cost-sharing religious-groups-in-schools anti-satellite-test-ban aid-to-nicaraguan-contras mx-missile immigration synfuels-corporation-cutback education-spending superfund-right-to-sue crime duty-free-exports export-administration-act-south-africa Class	Clustered Instances 0 142 (33%) 1 13 (3%) 2 45 (11%) 3 188 (44%) 4 27 (6%) 5 12 (3%) Unclustered instances : 8 Class attribute: Class Classes to Clusters: 0 1 2 3 4 5 <-- assigned to cluster 4 12 43 187 8 11 | democrat 138 1 2 1 19 1 | republican Cluster 0 <-- republican Cluster 1 <-- No class Cluster 2 <-- No class Cluster 3 <-- democrat Cluster 4 <-- No class Cluster 5 <-- No class	
2	0.9 / 6	75 , 17.2414 %	Relation: vote Instances: 435 Attributes: 17 el-salvador-aid Ignored: handicapped-infants water-project-cost-sharing adoption-of-the-budget-resolution physician-fee-freeze religious-groups-in-schools anti-satellite-test-ban aid-to-nicaraguan-contras mx-missile immigration synfuels-corporation-cutback education-spending superfund-right-to-sue crime duty-free-exports export-administration-act-south-africa Class	Clustered Instances 0 227 (52%) 1 208 (48%) Class attribute: Class Classes to Clusters: 0 1 <-- assigned to cluster 67 200 | democrat 160 8 | republican Cluster 0 <-- republican Cluster 1 <-- democrat	

- **<u>Observations :</u>**

- Here the attributes adoption of the budget resolution, religious groups in school, anti satellite test ban, crime, physician fee freeze and duty free exports were considered for the experimentation and the rest were ignored.
- Incase of DBSCAN, the number of incorrectly clustered instances increased from 95 to 185 and the number of clusters generated was 16.
- In the second case, attributes adoption of the budget resolution, physician fee freeze and el-salvador-aid were considered for the experimentation and the rest were ignored.
- The number of incorrectly clustered instances increased from 95 to 102 and the number of clusters generated was 6.
- In the third case only the attribute el-salvador-aid was considered.
- The number of incorrectly clustered instances decreased from 95 to 75 and the number of clusters generated was 2.
- From the above experimented results we can say el-salvador-aid is the important attribute in vote dataset.

Conclusion :

- When we considered KMeans and DBSCAN algorithms we found that KMeans performs better than DBSCAN in terms of incorrectly classified instances.
- When the dataset get denser the performance increases for DBSCAN algorithm is very higher when compared with KMeans. This could be because DBSCAN algorithm is designed for highly densed datasets while KMeans for Simpler datasets.
- In KMeans algorithm when the number of clusters increases the value of SSE decreases, giving better results.For a given value of K, the performance increase as the number of iterations increase. This is because as the number of iterations increase the data gets more refined. However very low K value do not yield good results.
- For DBSCAN, as the value of epsilon decreases the number of incorrectly clustered instances also decreases .Here for a given value of epsilon and as the minpoints value increase we get better results.
- When we ignore certain attributes and if the ignored attributes includes only worse attributes and does not include any best attributes then the SSE decreases giving a better result.
- On the other hand if the ignored attributes include the best attributes as then the results can be negative.
- Hence while evaluating the performance of these two algorithms on other data sets we need to consider the fact that DBSCAN is designed for densely datasets. It performs better for large data sets that are highly densed where as KMeans algorithm performs better for Simpler data sets.
- For example, when considered labor and diabetes data set :

Dataset	**Incorrectly clustered Instances**	**Incorrectly clustered Instances**
	KMeans	**DBSCAN**
Labor	13, 22.807%	Problem Evaluating Cluster : 0
Diabetes	255, 33.2031%	268, 34.8958%

- From the above table we can infer that SimpleKMeans performs better on labor data set as it has less number of incorrectly clustered instances.
- For Diabetes data set, again SimpleKMeans performs better.