Clustering Analysis Report

SOFTWARE USED

Weka is a collection of machine learning algorithms for data mining tasks. It contains tools for data preparation, classification, regression, clustering, association rules mining, and visualization.Weka is open source software issued under the **GNU General Public License**.

DATASET USED

I have used Pima Indians Diabetes Database as an arff file. The dataset has patients which are all females above the age of 21. The dataset shows weather they are diabetes positive or negative.

Number of Instances: 768

Number of Attributes: 8 plus class

For Each Attribute: (all numeric-valued)
1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

It is present at this url.

PREPROCESSING

I used Kmeans and Density based clustering. Kmeans is nominal towards noise and density based clustering is also sensitive towards noise. Thus I removed noise using the unsupervised filter known as “InterquartileRange”.

Now the dataset have 2 additional attributes noise and outlier. Then I removed the noise by using unsupervised filter known as “RemoveByValue”. Thus eliminating all the noise.

Before Preprocessing –

```
% Relabeled values in attribute 'class'
%     From: 0                       To: tested_negative
%     From: 1                       To: tested_positive
%
@relation pima_diabetes
@attribute 'preg' numeric
@attribute 'plas' numeric
@attribute 'pres' numeric
@attribute 'skin' numeric
@attribute 'insu' numeric
@attribute 'mass' numeric
@attribute 'pedi' numeric
@attribute 'age' numeric
@attribute 'class' { tested_negative, tested_positive}
@data
6,148,72,35,0,33.6,0.627,50,tested_positive
1,85,66,29,0,26.6,0.351,31,tested_negative
8,183,64,0,0,23.3,0.672,32,tested_positive
1,89,66,23,94,28.1,0.167,21,tested_negative
0,137,40,35,168,43.1,2.288,33,tested_positive
5,116,74,0,0,25.6,0.201,30,tested_negative
3,78,50,32,88,31,0.248,26,tested_positive
10,115,0,0,0,35.3,0.134,29,tested_negative
2,197,70,45,543,30.5,0.158,53,tested_positive
8,125,96,0,0,0,0.232,54,tested_positive
4,110,92,0,0,37.6,0.191,30,tested_negative
10,168,74,0,0,38,0.537,34,tested_positive
10,139,80,0,0,27.1,1.441,57,tested_negative
1,189,60,23,846,30.1,0.398,59,tested_positive
```

After Preprocessing –

```
@relation pima_diabetes-
weka.filters.unsupervised.attribute.InterquartileRange-Rfirst-
last-O3.0-E6.0-
weka.filters.unsupervised.instance.RemoveWithValues-S0.0-C10-
Llast

@attribute preg numeric
@attribute plas numeric
@attribute pres numeric
@attribute skin numeric
@attribute insu numeric
@attribute mass numeric
@attribute pedi numeric
@attribute age numeric
@attribute class {tested_negative,tested_positive}
@attribute Outlier {no,yes}
@attribute ExtremeValue {no,yes}

@data
6,148,72,35,0,33.6,0.627,50,tested_positive,no,no
1,85,66,29,0,26.6,0.351,31,tested_negative,no,no
8,183,64,0,0,23.3,0.672,32,tested_positive,no,no
1,89,66,23,94,28.1,0.167,21,tested_negative,no,no
5,116,74,0,0,25.6,0.201,30,tested_negative,no,no
3,78,50,32,88,31,0.248,26,tested_positive,no,no
8,125,96,0,0,0,0.232,54,tested_positive,no,no
4,110,92,0,0,37.6,0.191,30,tested_negative,no,no
10,168,74,0,0,38,0.537,34,tested_positive,no,no
10,139,80,0,0,27.1,1.441,57,tested_negative,no,no
5,166,72,19,175,25.8,0.587,51,tested_positive,no,no
```

PRELIMINARY ANALYSIS

I then did preliminary analysis using unsupervised filter attribute AddCluster. I performed Kmeans analysis on the dataset. The class predicted by it was more accurate after removing noise.

Preliminary Analysis Dataset –

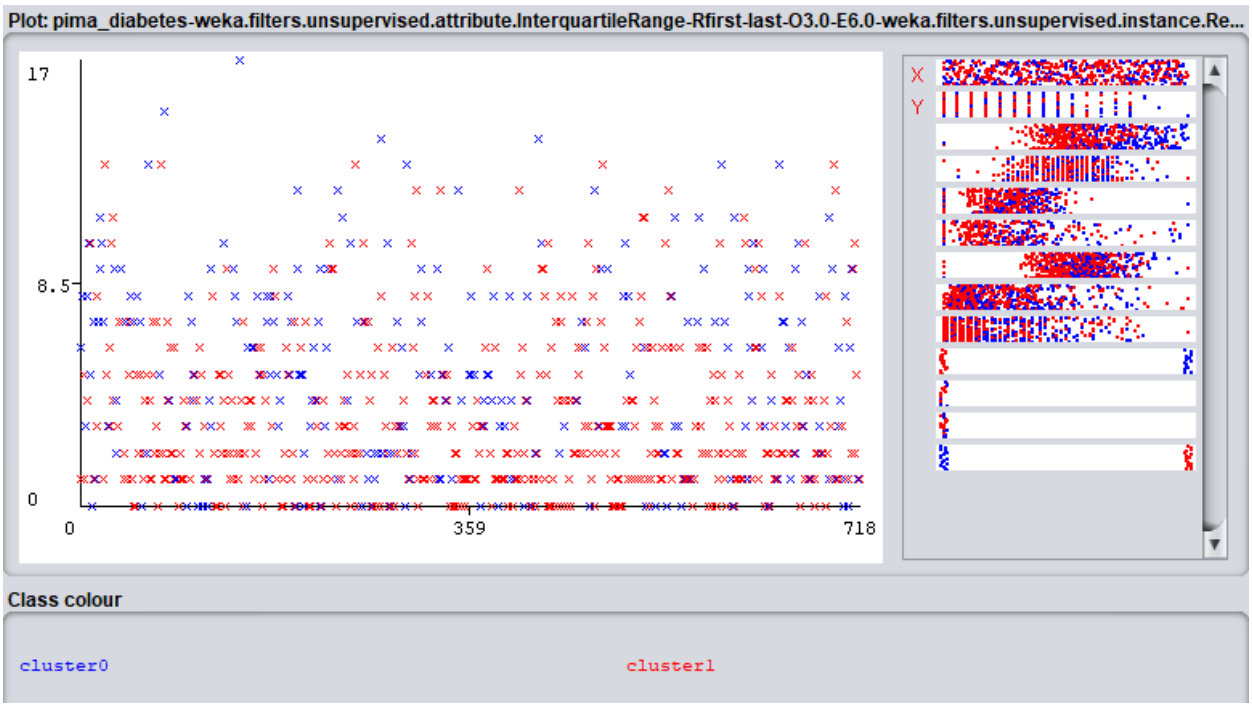
```
@relation 'pima_diabetes-
weka.filters.unsupervised.attribute.InterquartileRange-Rfirst-
last-O3.0-E6.0-
weka.filters.unsupervised.instance.RemoveWithValues-S0.0-C10-
Llast-weka.filters.unsupervised.attribute.AddCluster-
Wweka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -
periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N
2 -A \"weka.core.EuclideanDistance -R first-last\" -I 500 -num-
slots 1 -S 10'

@attribute preg numeric
@attribute plas numeric
@attribute pres numeric
@attribute skin numeric
@attribute insu numeric
@attribute mass numeric
@attribute pedi numeric
@attribute age numeric
@attribute class {tested_negative,tested_positive}
@attribute Outlier {no,yes}
@attribute ExtremeValue {no,yes}
@attribute cluster {cluster1,cluster2}

@data
6,148,72,35,0,33.6,0.627,50,tested_positive,no,no,cluster1
1,85,66,29,0,26.6,0.351,31,tested_negative,no,no,cluster2
8,183,64,0,0,23.3,0.672,32,tested_positive,no,no,cluster1
1,89,66,23,94,28.1,0.167,21,tested_negative,no,no,cluster2
5,116,74,0,0,25.6,0.201,30,tested_negative,no,no,cluster2
3,78,50,32,88,31,0.248,26,tested_positive,no,no,cluster1
```

KMEANS CLUSTERING

I used kmeans clustering by ignoring the class attribute.

Here are the Results-

```
Number of iterations: 3
Within cluster sum of squared errors: 155.85979691097486

Initial starting points (random):

Cluster 0: 5,139,80,35,160,31.6,0.361,25,tested_positive,no,no
Cluster 1: 5,103,108,37,0,39.2,0.305,65,tested_negative,no,no

Missing values globally replaced with mean/mode

Final cluster centroids:
                                        Cluster#
Attribute               Full Data             0                1
                          (719.0)         (242.0)          (477.0)
=============================================================================
preg                       3.8943          5.0579           3.304
plas                     120.2017        140.4917        109.9078
pres                      72.3853         75.5331         70.7883
skin                       21.146         22.7727         20.3208
insu                       77.096          95.062         67.9811
mass                      32.1029         34.9285         30.6694
pedi                       0.4623          0.5347          0.4256
age                       33.3394         37.5165         31.2201
class           tested_negative tested_positive tested_negative
Outlier                       no              no               no
ExtremeValue                  no              no               no



Time taken to build model (full training data) : 0 seconds

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

Clustered Instances

0      242 ( 34%)
1      477 ( 66%)
```



EVALUATING KMEANS CLUSTERING

I did classes to cluster evaluation with “(Nom)class” attribute. I ignored the class attribute.

Here are the results –

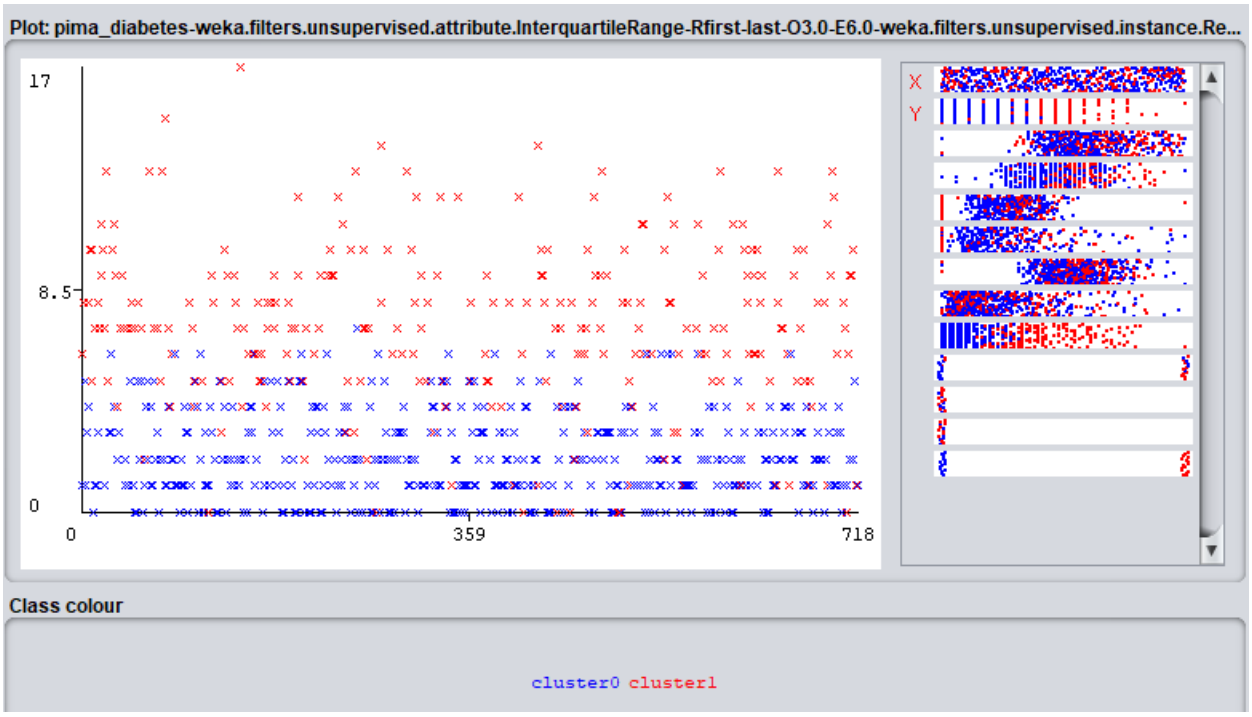
```
Time taken to build model (full training data) : 0.02 seconds

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

Clustered Instances

0      472 ( 66%)
1      247 ( 34%)


Class attribute: class
Classes to Clusters:

   0   1  <-- assigned to cluster
 355 122 | tested_negative
 117 125 | tested_positive

Cluster 0 <-- tested_negative
Cluster 1 <-- tested_positive

Incorrectly clustered instances :       239.0    33.2406 %
```

DENSITY BASED CLUSTERING

I used density based clustering on the dataset, because my assumption was that it would give better results then Kmeans because Kmeans only works well for globular clusters.

Here are the results –

```
Time taken to build model (full training data) : 0.02 seconds

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

Clustered Instances

0       454  ( 63%)
1       265  ( 37%)


Log likelihood: -28.46198
```



EVALUATING DENSITY BASED CLUSTERING

I did classes to cluster evaluation with “(Nom)class” attribute. I ignored the class attribute.

Here are the results –

```
Time taken to build model (full training data) : 0.03 seconds

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

Clustered Instances

0      454 ( 63%)
1      265 ( 37%)


Log likelihood: -28.46198


Class attribute: class
Classes to Clusters:

   0   1  <-- assigned to cluster
 348 129 | tested_negative
 106 136 | tested_positive

Cluster 0 <-- tested_negative
Cluster 1 <-- tested_positive

Incorrectly clustered instances :       235.0    32.6843 %
```

RESULT ANALYSIS

Both the algorithms performed as expected. The density based clustering performed a little better then Kmeans because the data was not globular. But density based cluster too deviated from good results because the density varied.

Here is the comparison –

For Kmeans –

Incorrectly clustered instances : 239.0 33.2406 %

For Density Based Clustering –

Incorrectly clustered instances : 235.0 32.6843 %