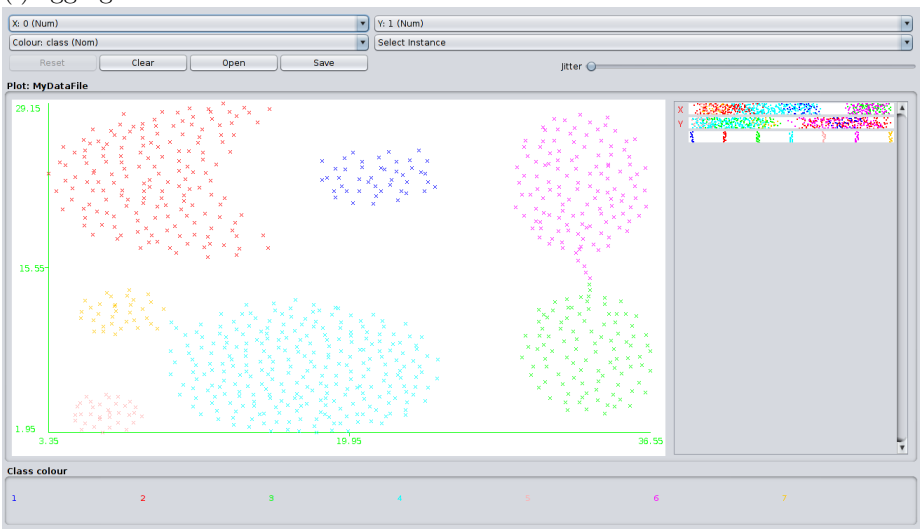
CS4011:Assignment 3

S Aakash CS15B060

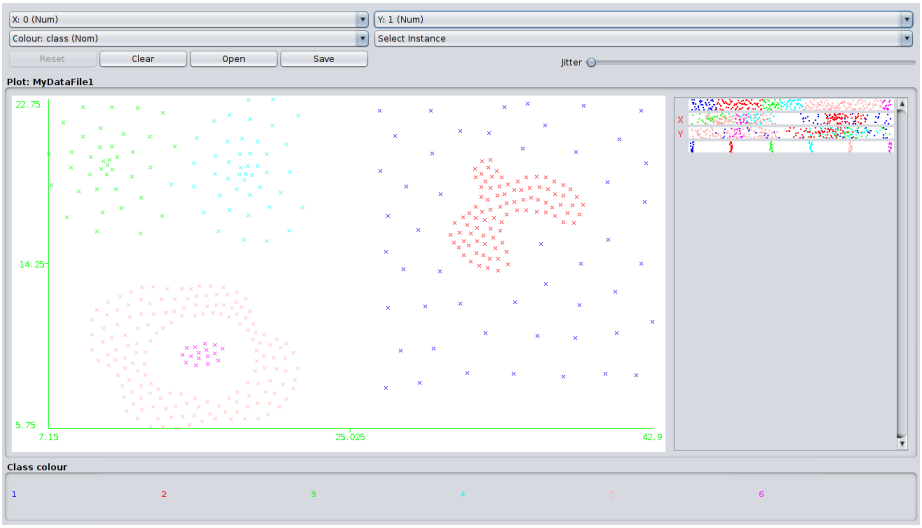
November 10, 2017

0.1 Question-1

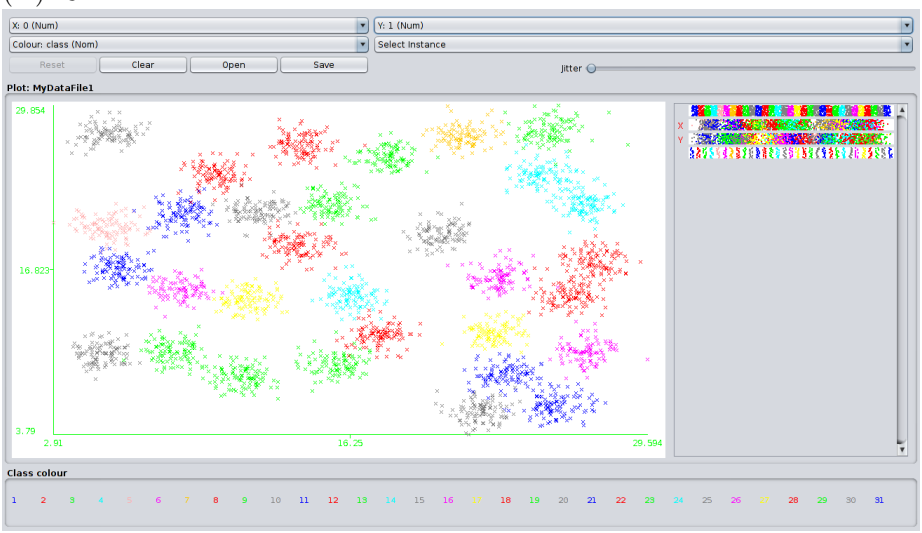
All the 8 datasets were converted to ARFF format named as filename1.arff .Now weka is used to visualise each of these datasets.Note that we need not think about class corresponding to data point in visualisation.(ie colour of cluster is not of concern right now)
(i)Aggregation:



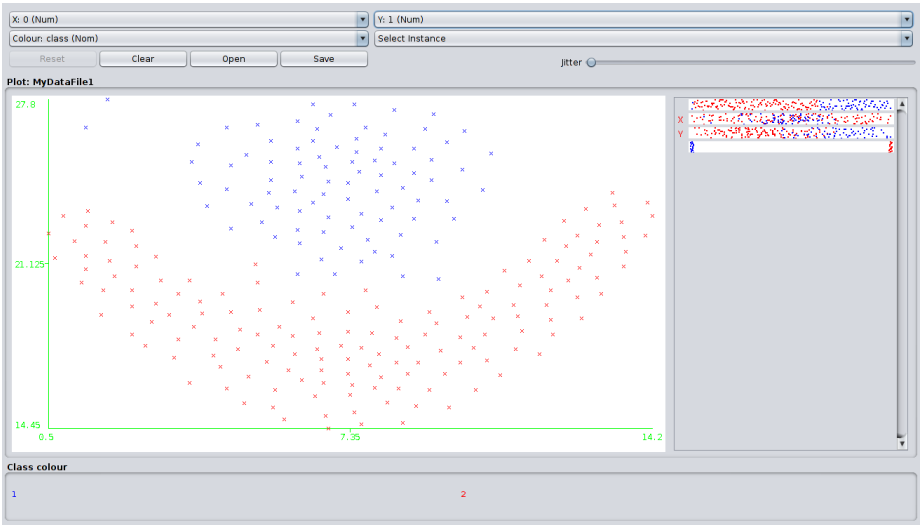
k means(k=6) could perform well provided we have a good initilisation of centroids. Hierarchical with single link could potentially merge the two classes on right and similarly two on left,whereas complete-link can handle situations better.DBScan may also result in similar situations(ofcourse we need to tune eps,min.pts carefully to avoid this situation).
(ii)Compound:



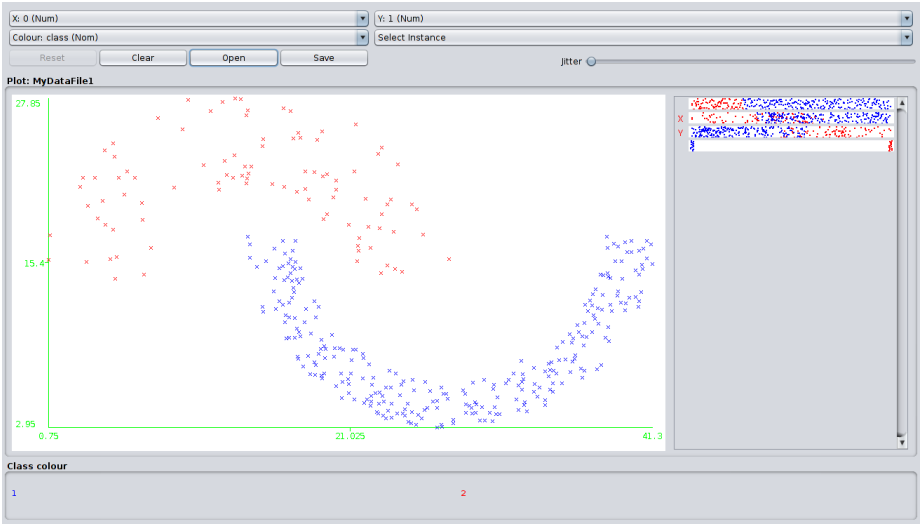
K-means will not work well in this situation.Heirarchial could possibly handle it,as the visualisation says that cluster within cluster is preferrable.Single link performs better,except(it merges 2 classes on top-left)which will be handled by multi-link.DBScan is not preferrable,as the clusters are of different densities.
(iii)D31:



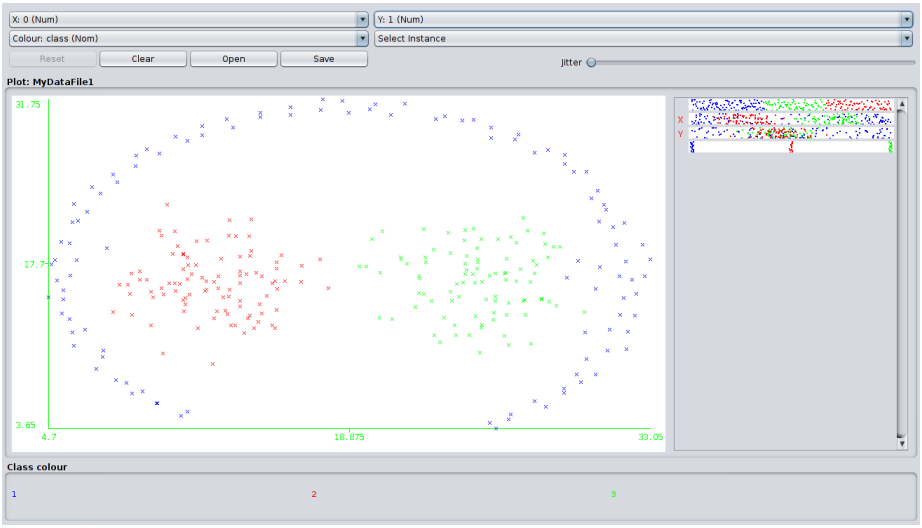
Here K-means(k=31,maybe) could work out well,provided good initialisation but even DB-Scan could perform well(note all clusters look like they have uniform densities).Hierarchial may not be as effective as both of these.
(iv)Flames:



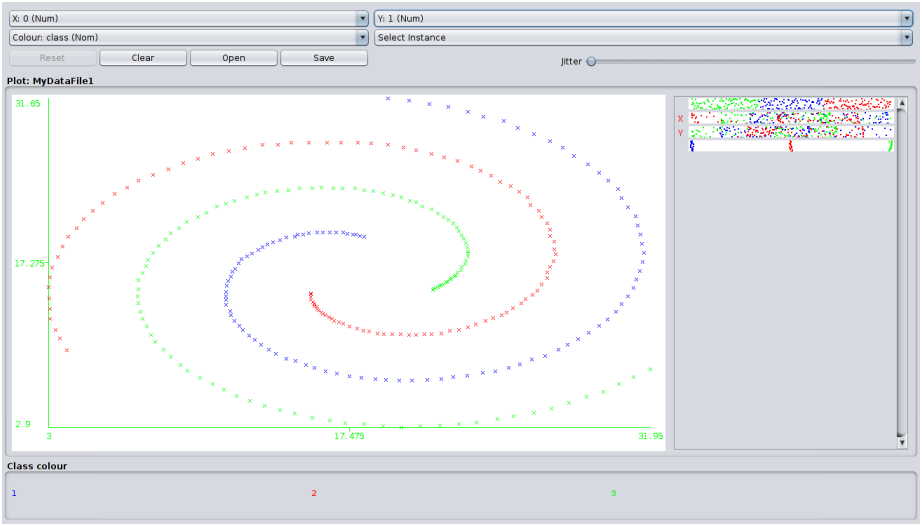
DBScan may not perform well(as the two clusters are kind of close enough to each other and it wont be able to demarcate between two.).Hierarchical with single link/complete link also wont be good,maybe avg. link could be good.K-means(k=2) may not perform well as it generaly produces a convex boundry.
(v)Jain:



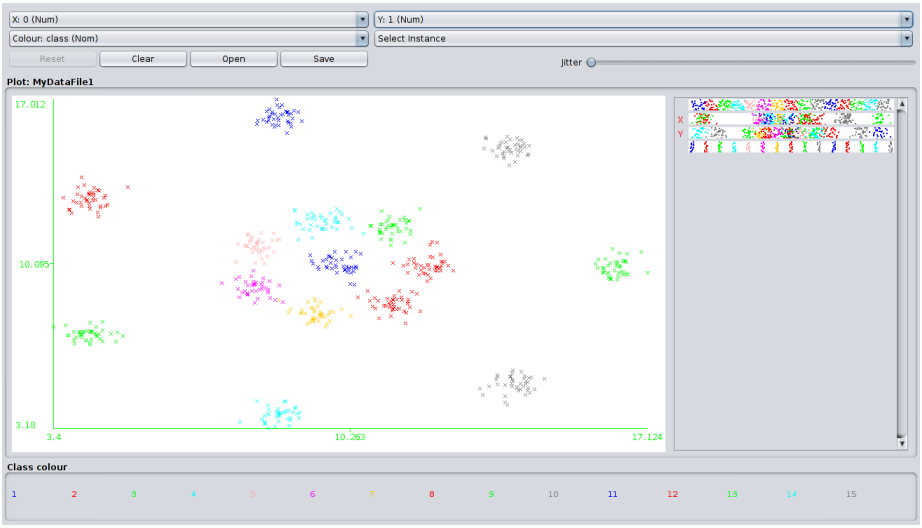
It is one of the best examples of DBScan,where two required clusters will be obtained by algo(Note that we need to tune parameters properly).k-means may turn out to be bad.Hierarchial(complete link,num clusters=2) could perform fairly.
(vi)Path-based:



DBScan performs best,as essentially traversal along densely connected regions will yield us required clusters.k-means will turn out to be bad,Hierarchical with complete link will perform better than single-link.
(vii)Spiral:



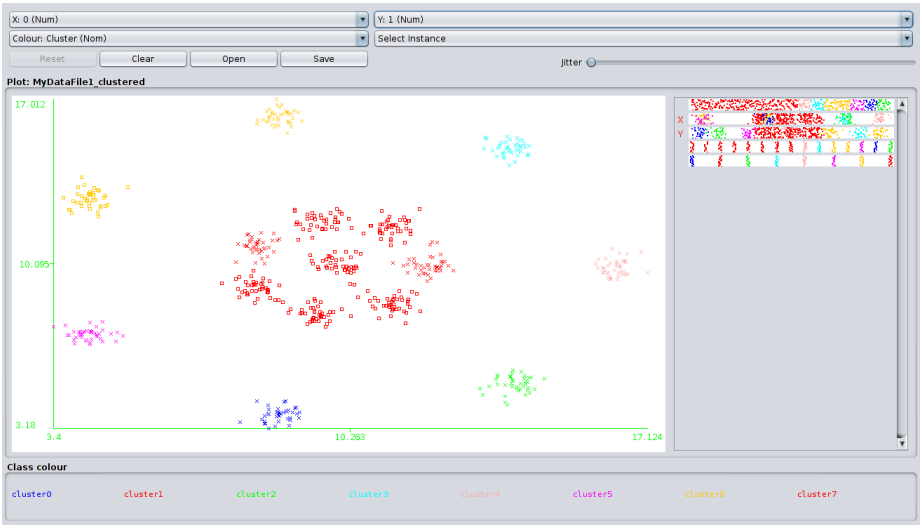
Again a best example of DBScan,traversal along densely connected path gives u clusters.K-means will perform worse and Hierarchical clustering with complete-link will also perform good,with single link,it does average.
(viii)R15:



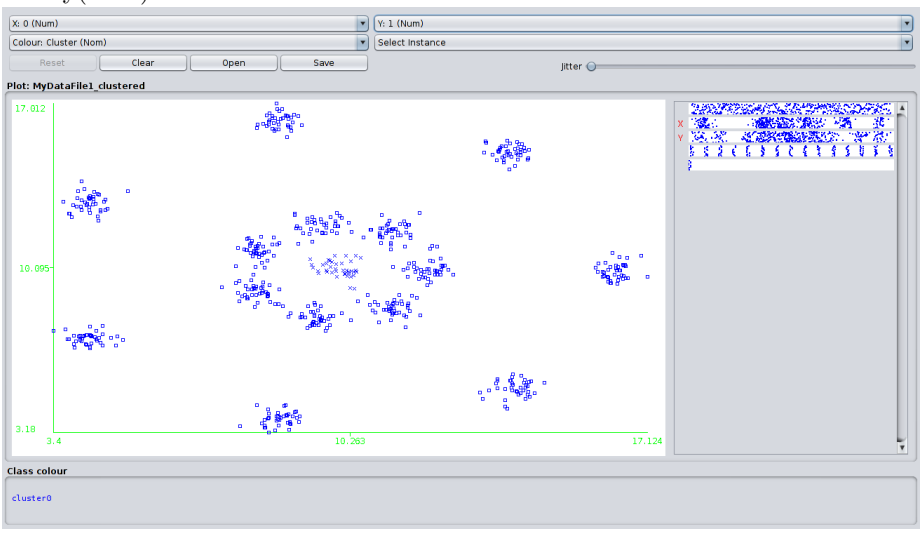
k-means with k=15 will do good,intution is explained in next subdivision.DB scan could do,but careless parameter setting will make the middle part as a single cluster.Heirarchical cluster(single and complete) may also perform well,but again it depends on how many clusters we want.

0.1.1 K-means with R15

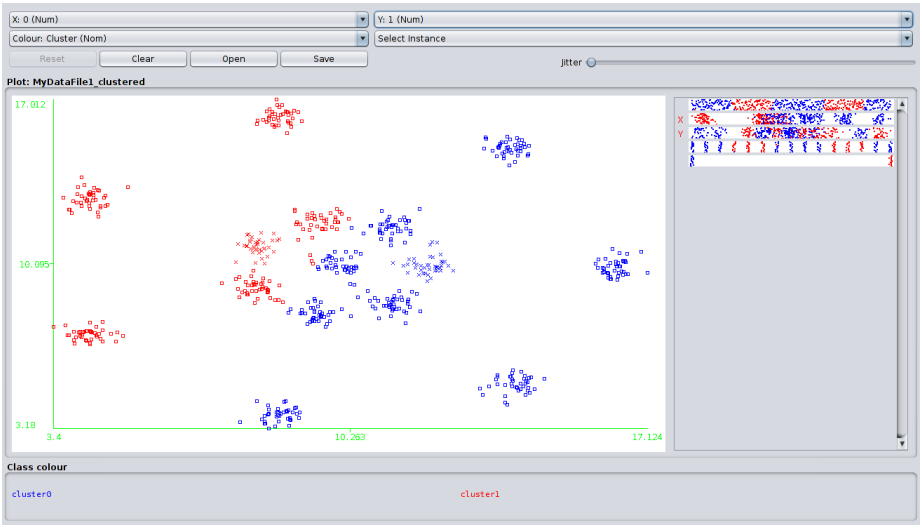
Random initialisation,Farthest distance initialisatios were tried out.Farthest distance was better.Here is the preformance of K-means with R15,where k=8:
Incorrectly clustered instances : 280.0
Purity=53.33%
Intutively we can say that k=8 will create 1-middle cluster,7-other clusters surrounding it,provided the initialisation is good enough and kmeans is performed multiple times.So Weka also gives us the same expected output:



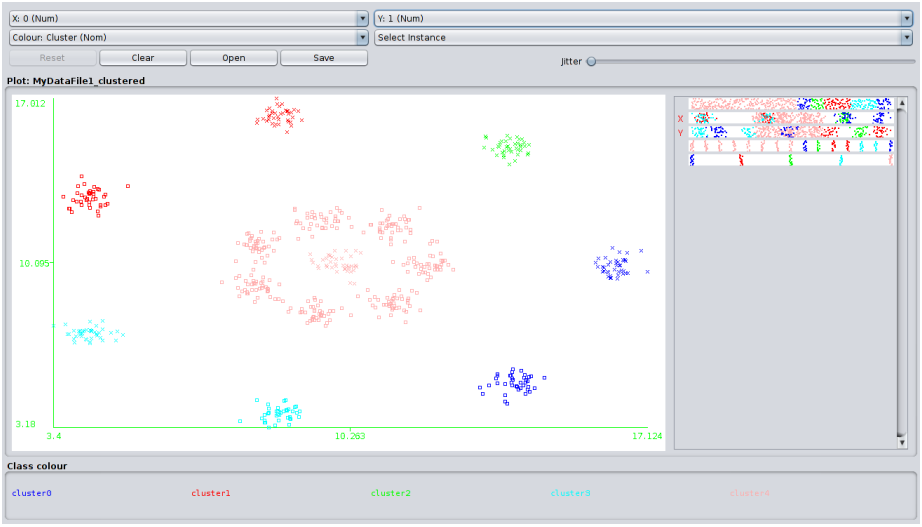
But purity is ofcourse quite less as we didn't really look into class labels while applying clustering.
(i)k=1:
A single cluster:
Incorrectly clustered instances : 560.0
Purity(*100)=6.33%



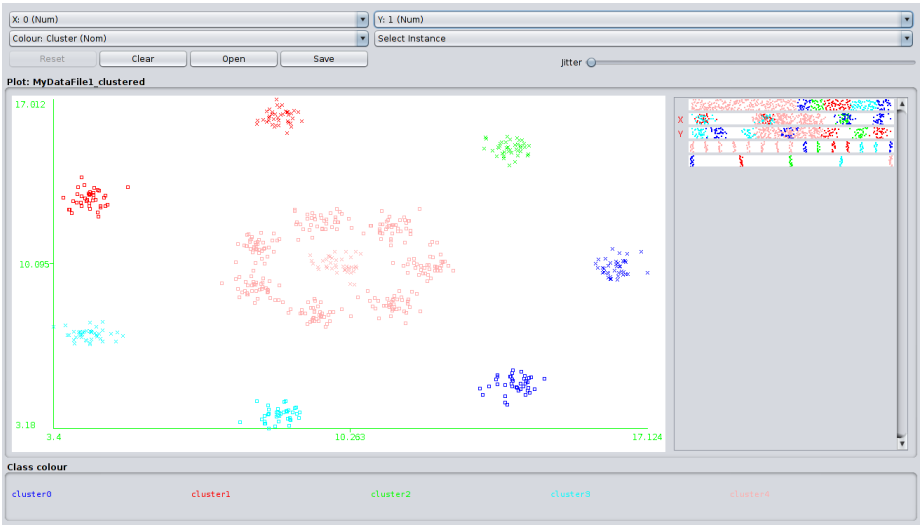
(ii)k=2:
two cluster:
Incorrectly clustered instances : 520.0
Purity(*100)=13.33%



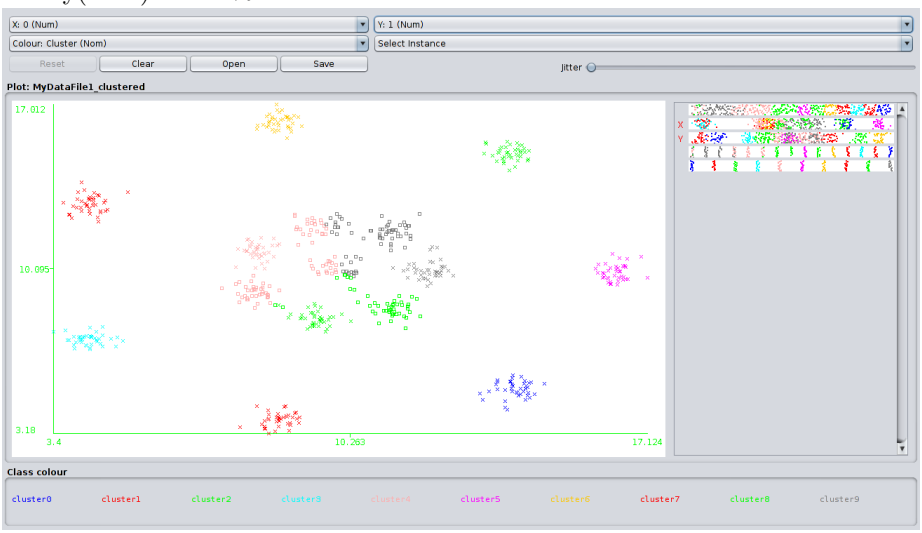
(iii)k=5:
Incorrectly clustered instances : 400.0
Purity(*100)=33.33%



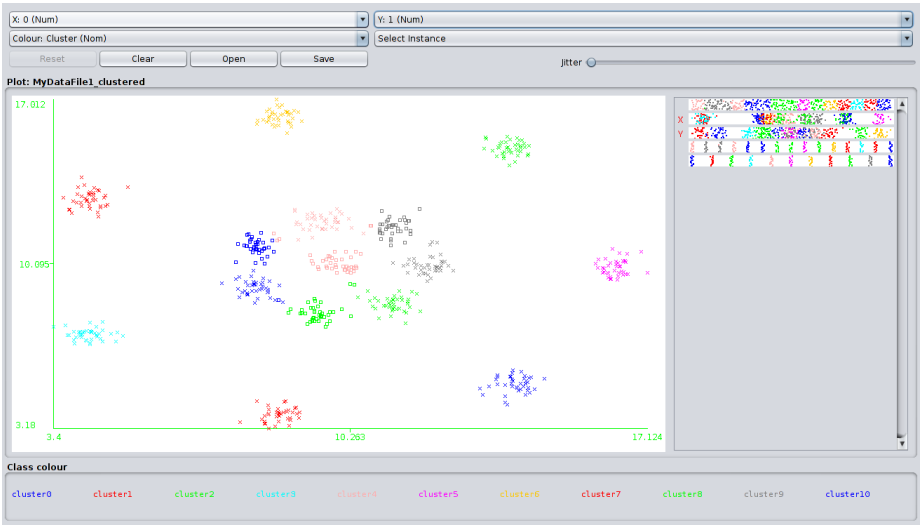
(iv)k=5:
Incorrectly clustered instances : 400.0
Purity(*100)=33.33%



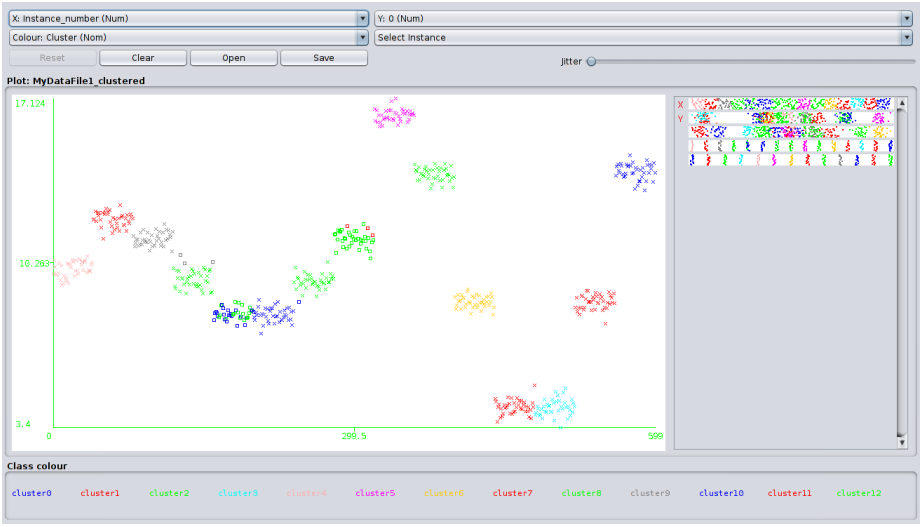
(v)k=10:
Incorrectly clustered instances : 200.0
Purity(*100)=66.66%



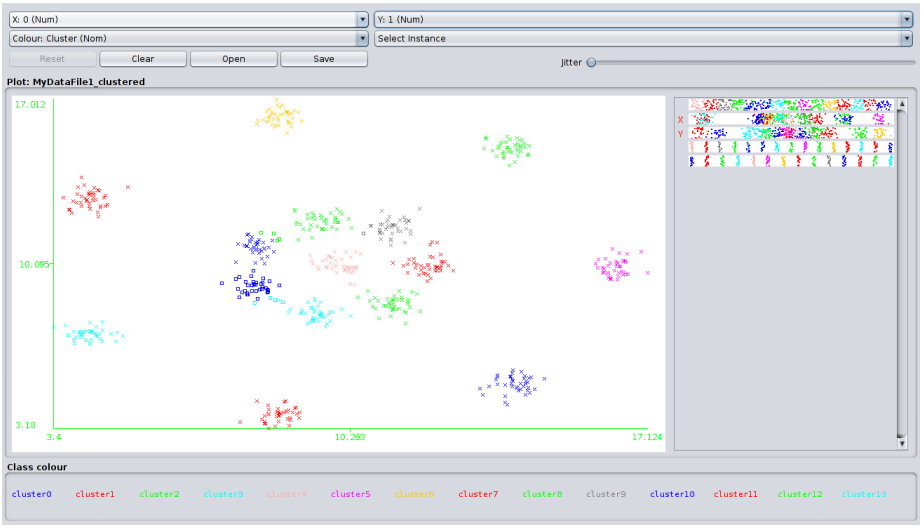
(vi)k=11:
Incorrectly clustered instances : 160.0
Purity(*100)=73.33%



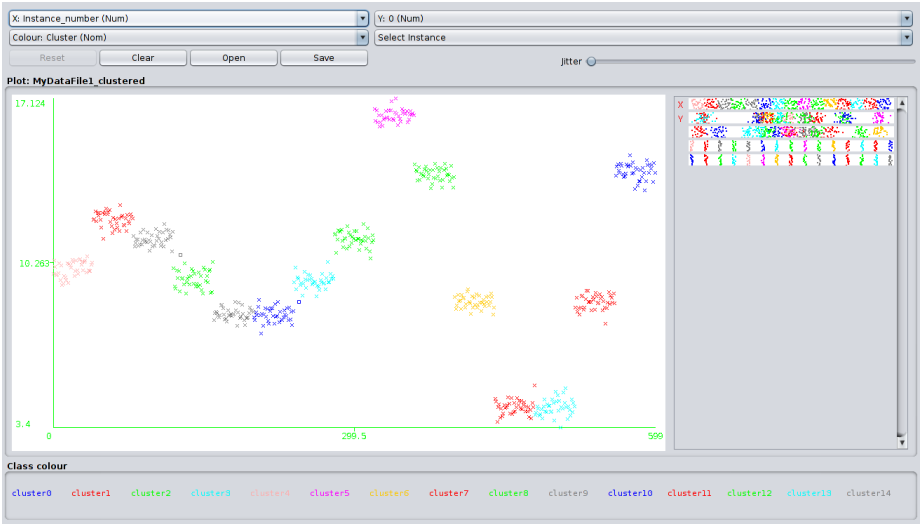
(vii)k=13:
Incorrectly clustered instances : 84.0
Purity(*100)=86.00%



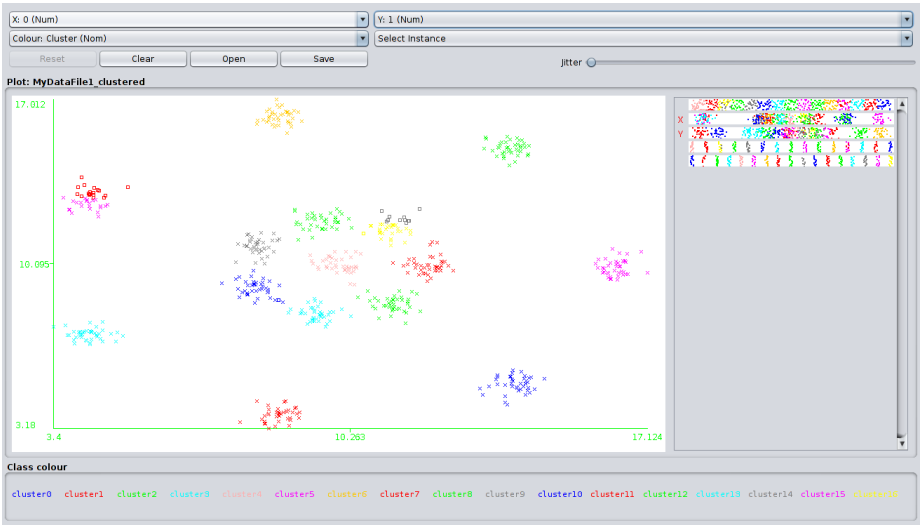
(viii)k=14:
Incorrectly clustered instances : 45.0
Purity(*100)=7.5%



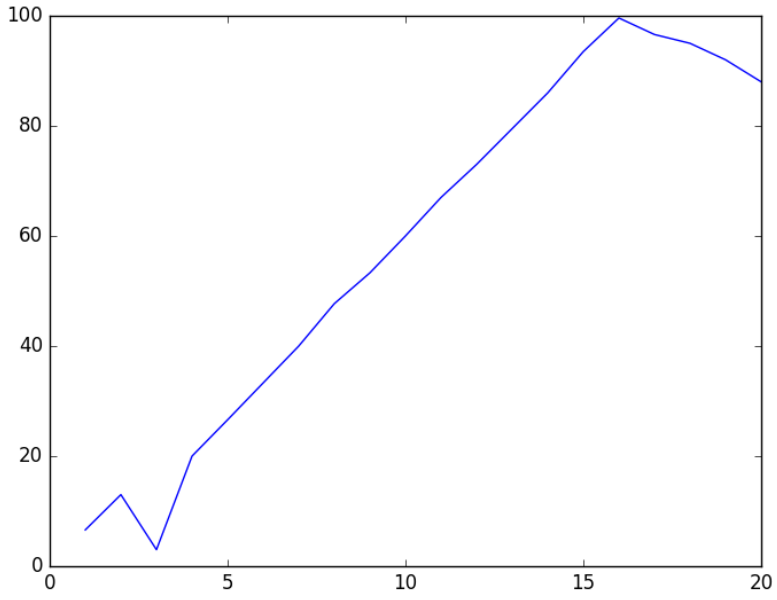
(ix)k=15:
Incorrectly clustered instances : 2.0
Purity(*100)=0.33%



(iii)k=17:
Incorrectly clustered instances : 31.0
Purity(*100)=5.16%



So k=15 is best in terms of purity.Intutively itself we can feel that in some sense(ie in unlabelled data),k=8,k=15 are quite good.Infact in k=8,we consider innermost region as a single cluster,whereas in k=15,the innermost one is divided into 8 different clusters.
Here is the plot(Purity(*100)-Y axis and K-X axis):

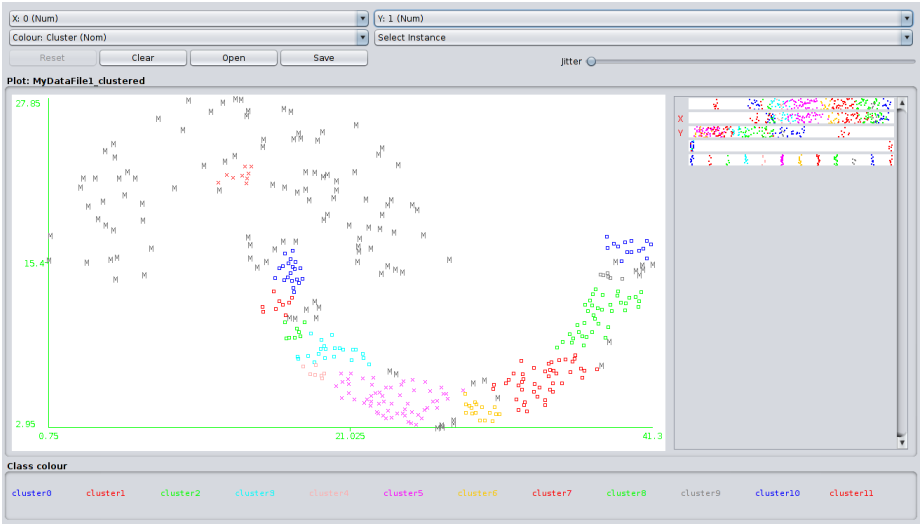


So as explained before k=15 is having a good performance.We can see that as k increases from 1 to 15,purity increases(It justifies our intutive understanding

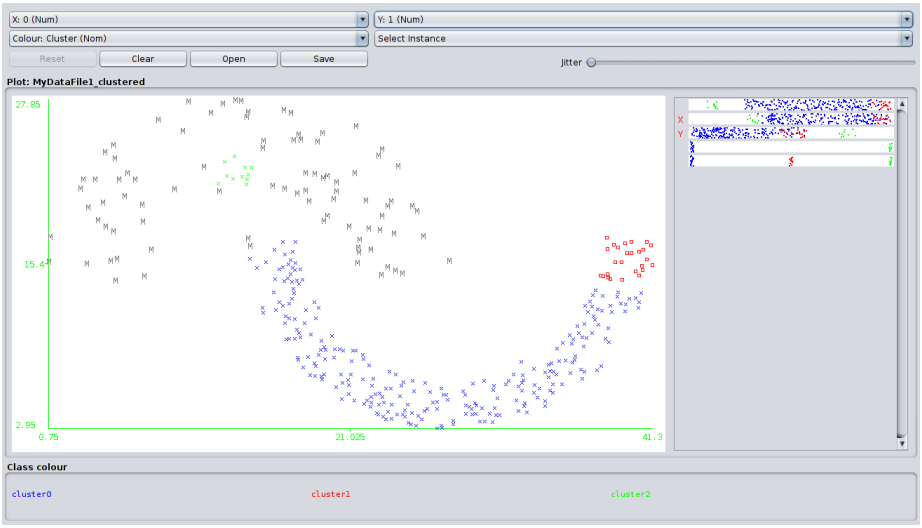
that as we approach k=15,we get better and better).Also once k increases and k≥15,since we already reached a best soln,any further increase in k just splits up the cluster further,leading to potential decrease in purity.

0.1.2 DB Scan with Jain Dataset

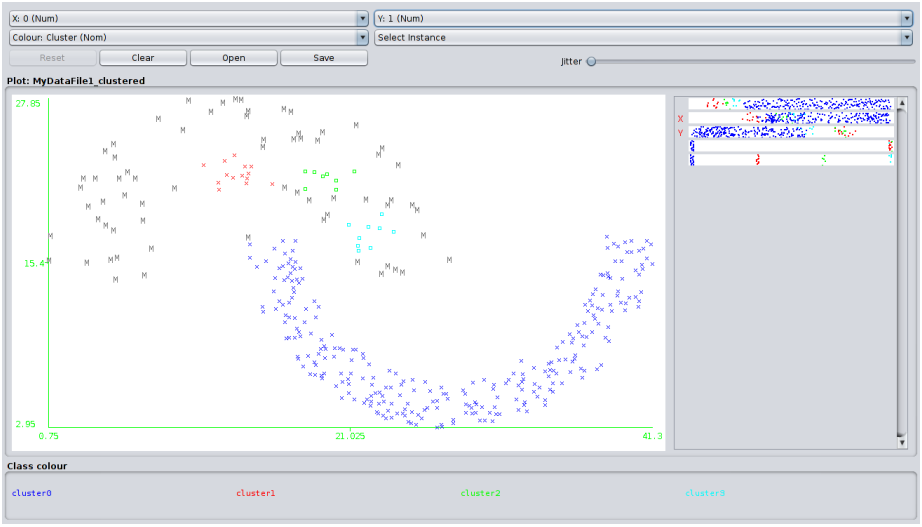
Basically min.pts and eps. determine(ie can strengthen/weaken) our definition of dense region.If eps. increases/decreases,keeping min.pts constant,then definition strength density decreases/increases respectively.Similarly If min.pts increases/decreases,keeping eps. constant,then definition/strength of density rich region decreases/increases respectively and no. of clusters decrease/increase respectively.
Effect of eps.(To observe this,lets keep min.pts=constant):
(i)eps.=0.03,min.pts=6:
Incorrectly clustered instances : 185.0
Purity(*100)=49.59%



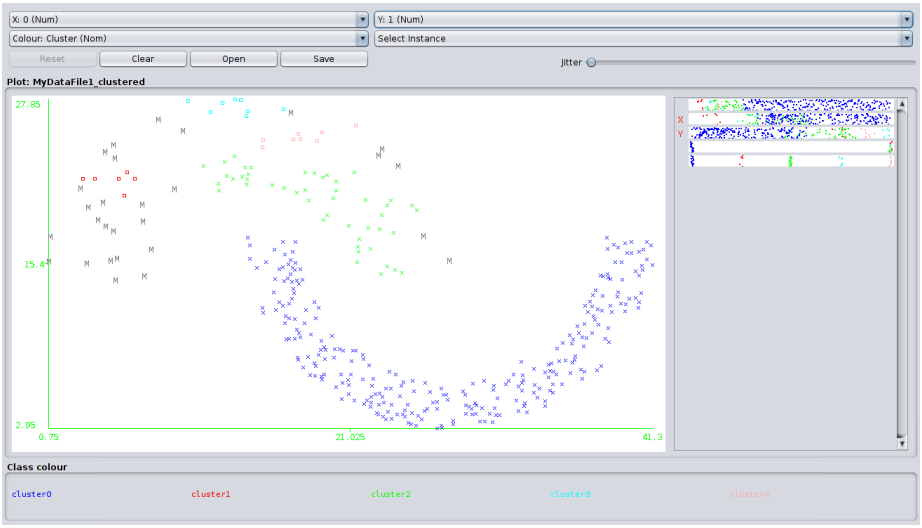
(ii)eps.=0.04,min.pts=6:
Incorrectly clustered instances : 26.0
Purity(*100)=6.97%



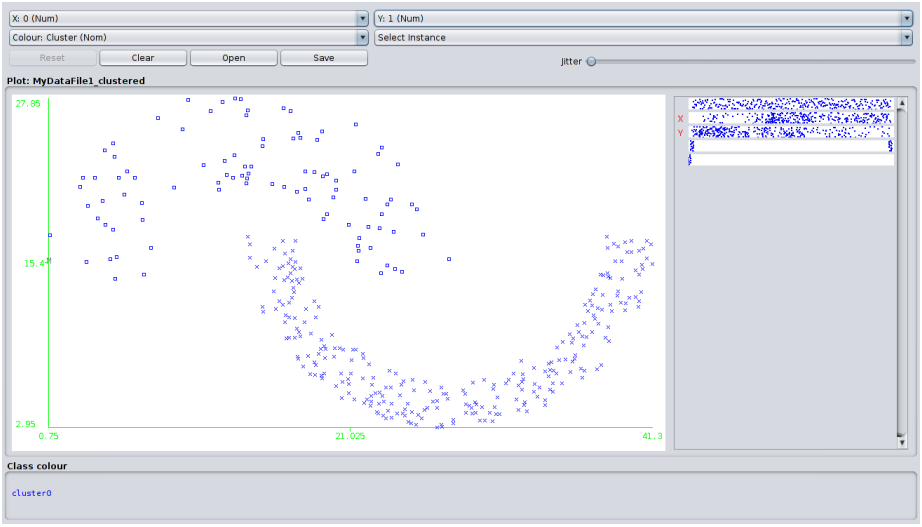
(iii)eps.=0.05,min.pts=6:
Incorrectly clustered instances : 17.0
Purity(*100)=4.55%



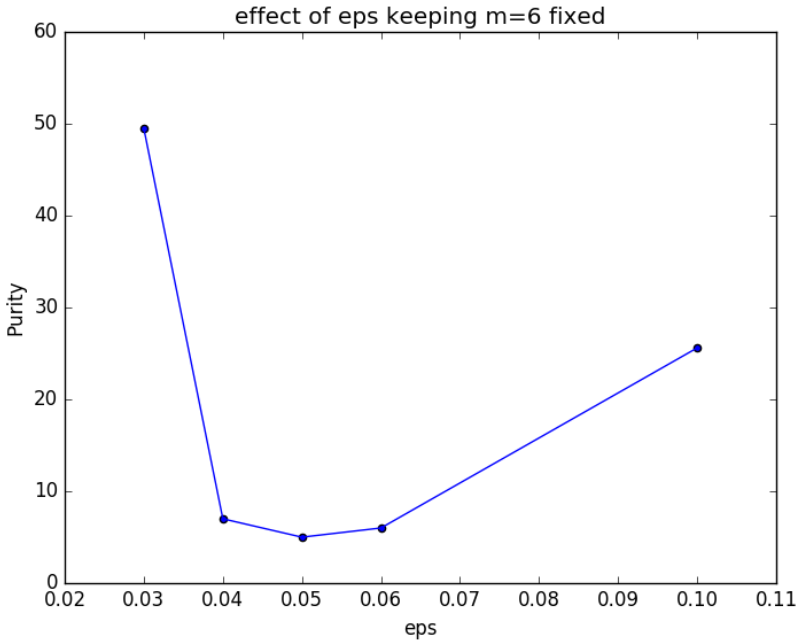
(iv)eps.=0.06,min.pts=6:
Incorrectly clustered instances : 22.0
Purity(*100)=5.89%



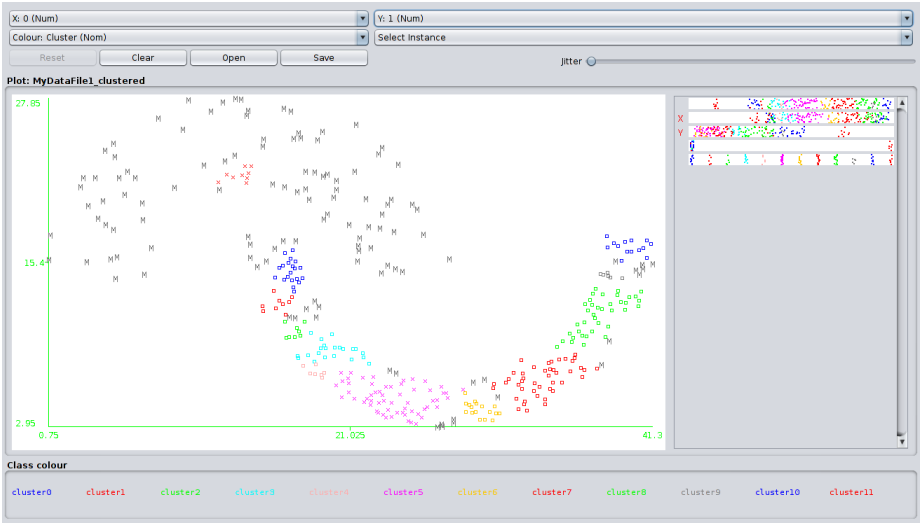
(v)eps.=0.1,min.pts=6:
Incorrectly clustered instances : 96.0
Purity(*100)=25.73%



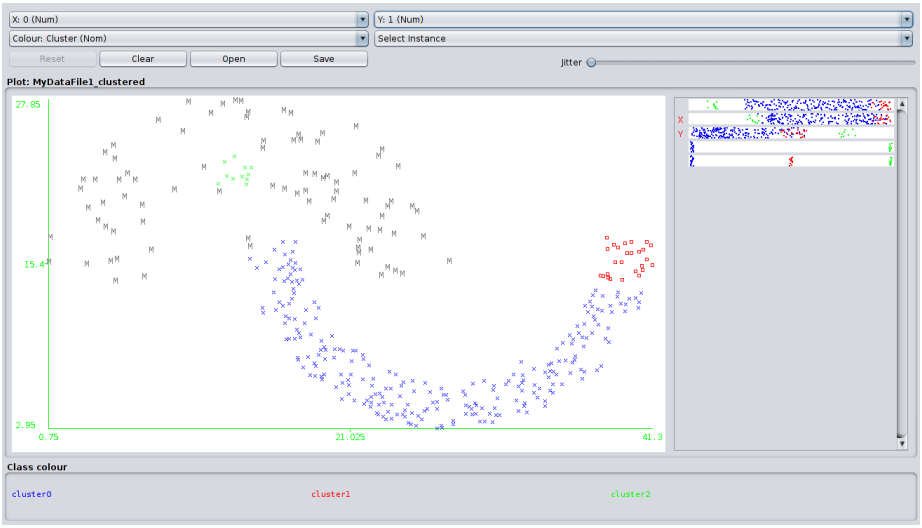
We can observe a beautiful thing.Basically the dataset is just 2 seperate(banana shaped curves) and we intuitively know that these 2 are potential to beome clusters in DB Scan.Now as eps increases from 0.01 to 0.06,the defn of density weakens and the lower region slowly meregs into single cluster.As we increase further,upper region also starts to merge ino single region



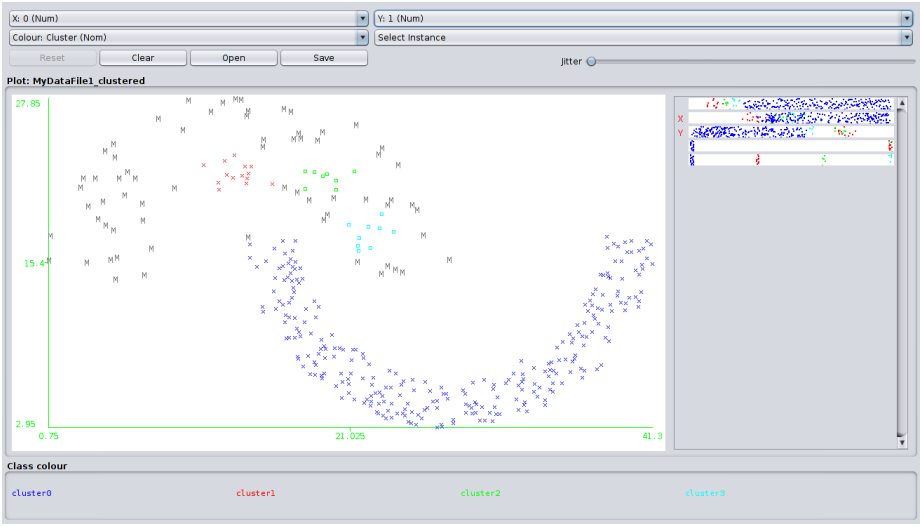
(ii)Effect of m keeping eps fixed:
(i)eps.=0.06,min.pts=10:
Incorrectly clustered instances : 185.0
Purity(*100)=49.59%



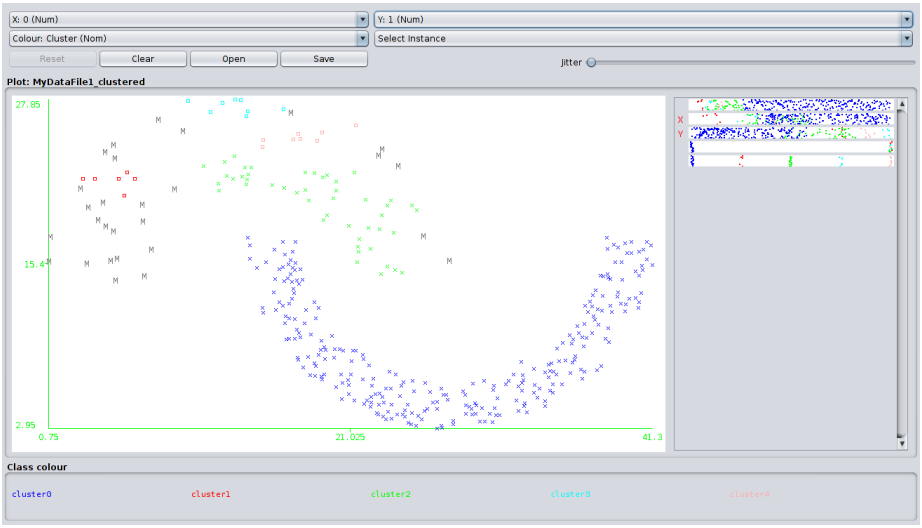
(ii)eps.=0.06,min.pts=6:
Incorrectly clustered instances : 26.0
Purity(*100)=6.97%



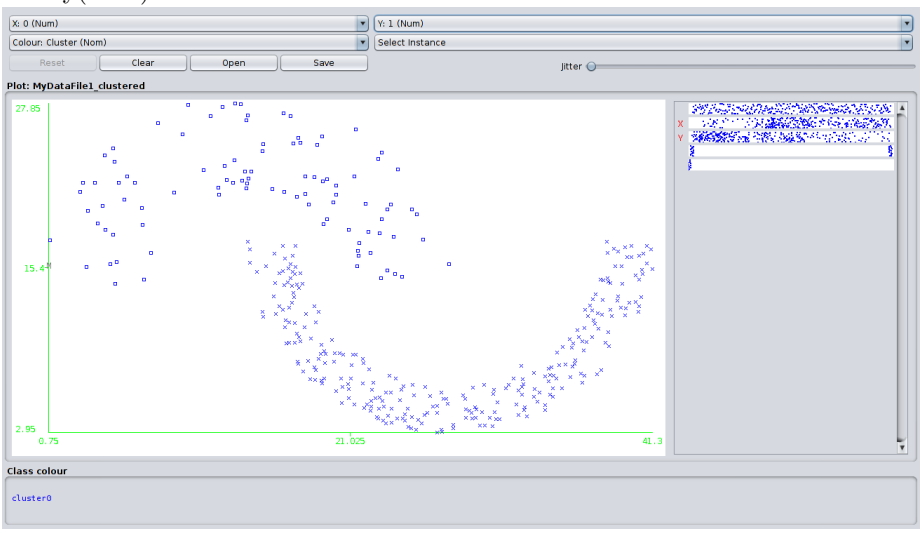
(iii)eps.=0.06,min.pts=5:
Incorrectly clustered instances : 17.0
Purity(*100)=4.55%



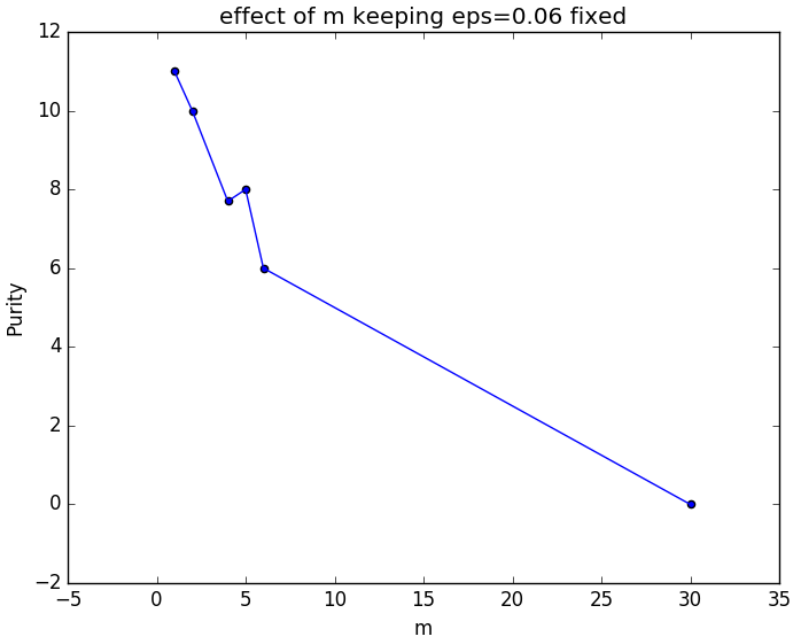
(iv)eps.=0.06,min.pts=4:
Incorrectly clustered instances : 22.0
Purity(*100)=5.89%



(v)eps.=0.06,min.pts=2:
Incorrectly clustered instances : 96.0
Purity(*100)=25.73%
(vi)eps.=0.06,min.pts=30:
Incorrectly clustered instances : 185.0
Purity(*100)=49.59%

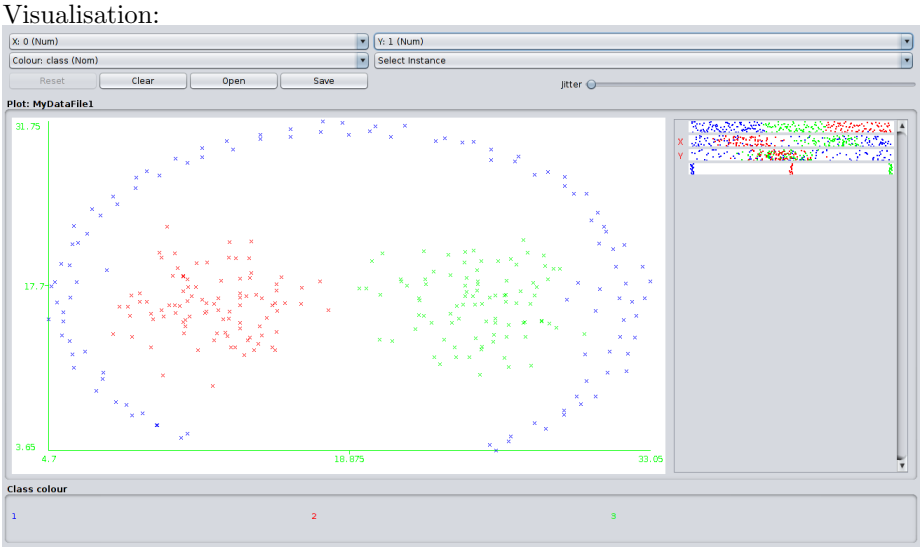


Graph showing effect of m:

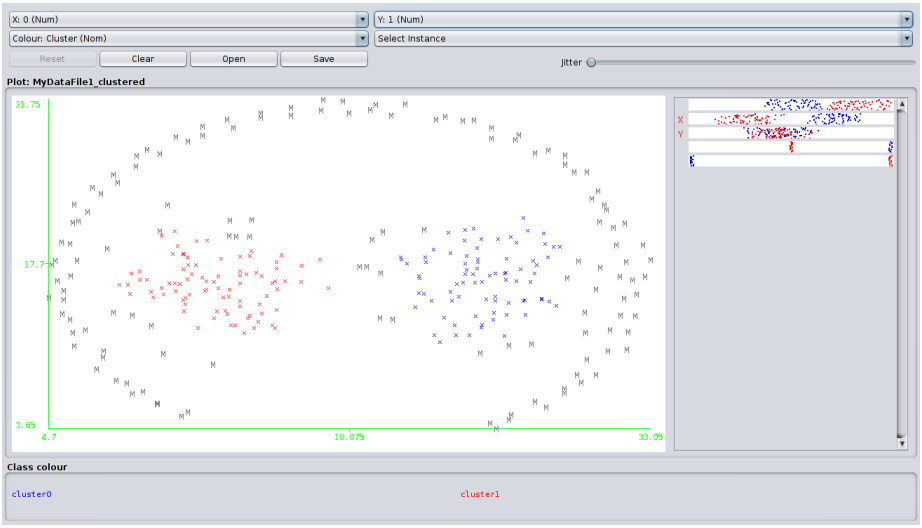


0.2 Path-based

Visualisation:



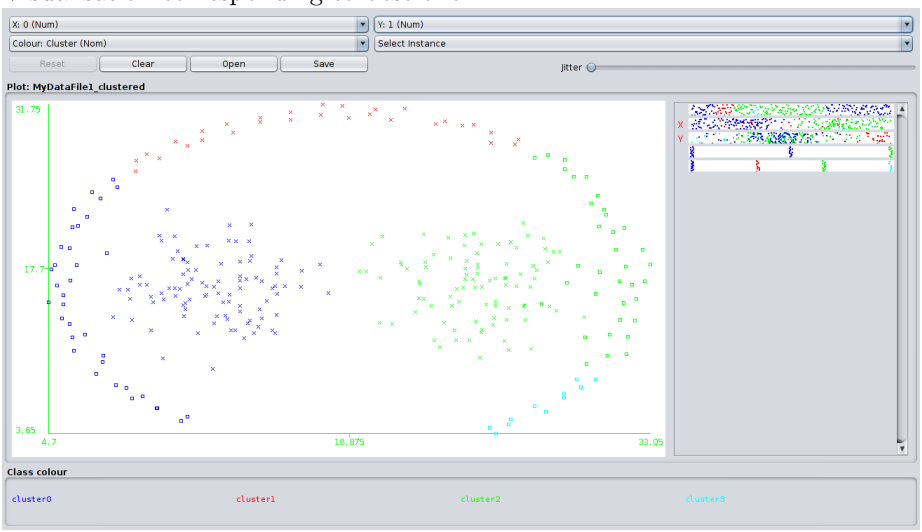
Best performance with DB Scan occured at:eps=0.06,m=5



Incorrectly clustered instances : 0.0
Purity(*100)=100.00%(of course many pts weren't in cluster,but by defn,purity=100)
Performances with following parameters for Hierarchial with Numclusters=3:

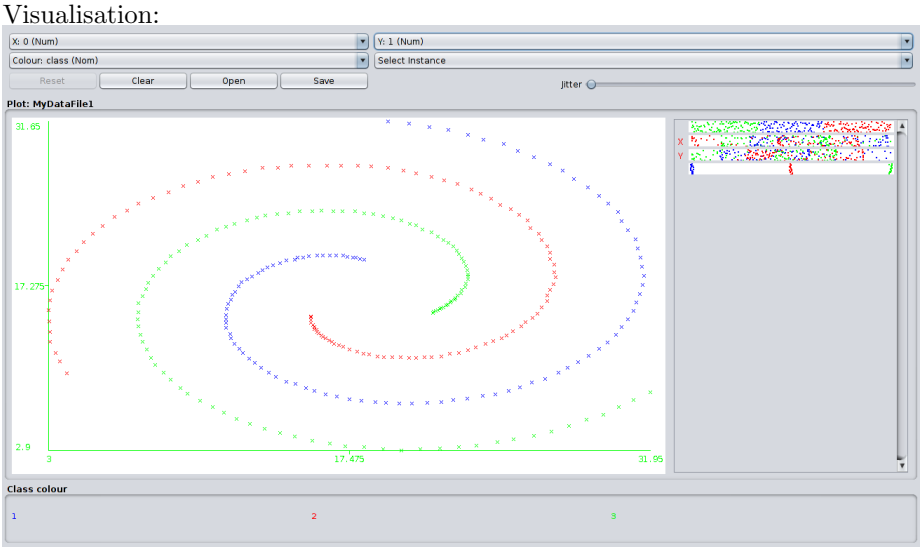
```
single link: Purity  37%
complete link:Purity  56%
average link:Purity 73 %
mean:Purity 70%
centroid: Purity 73.33 %
ward:Purity 70.66%
adjcomplete:Purity    64 %
neighbour join:Purity 36.66 %
best centroid:Purity 73.33%
```

Visualisation corresponding to best one:

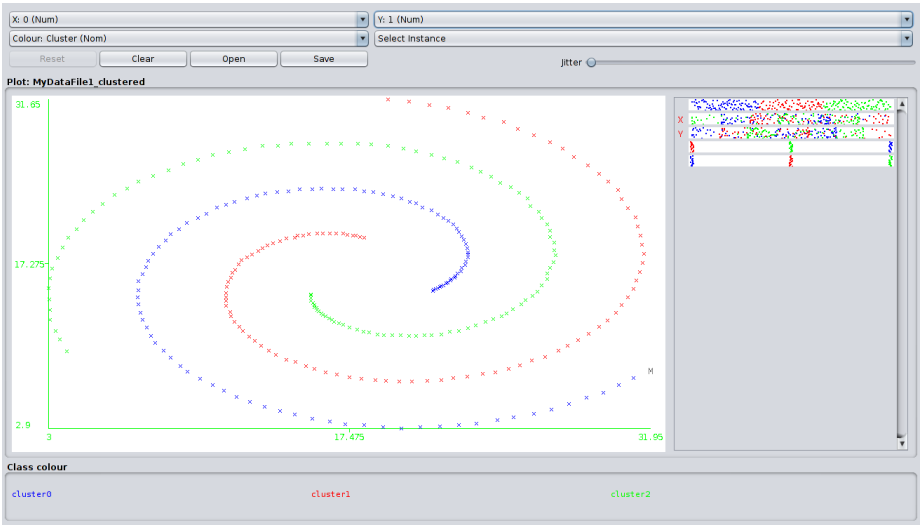


0.3 Spiral

Visualisation:



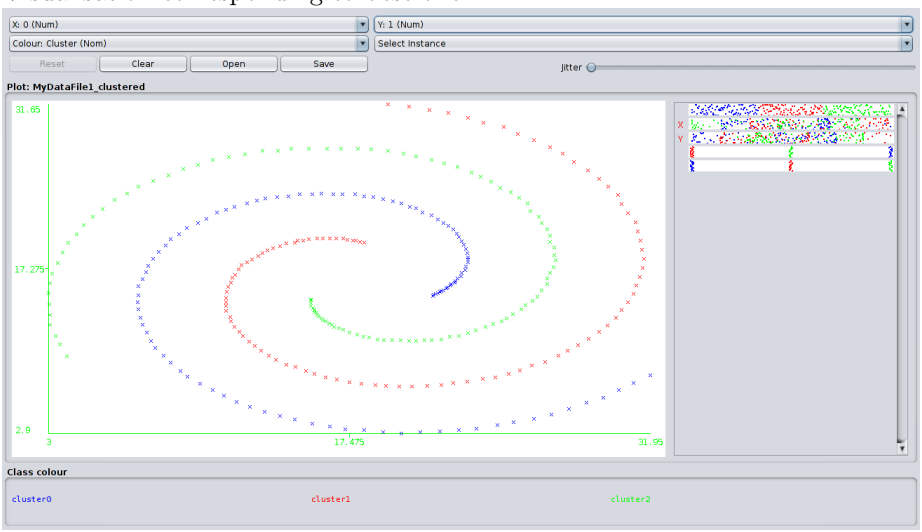
Best performance with DB Scan occured at:eps=0.1,m=5



Incorrectly clustered instances : 0.0
Purity(*100)=100.00%)
Performances with following parameters for Hierarchial with Numclusters=3:

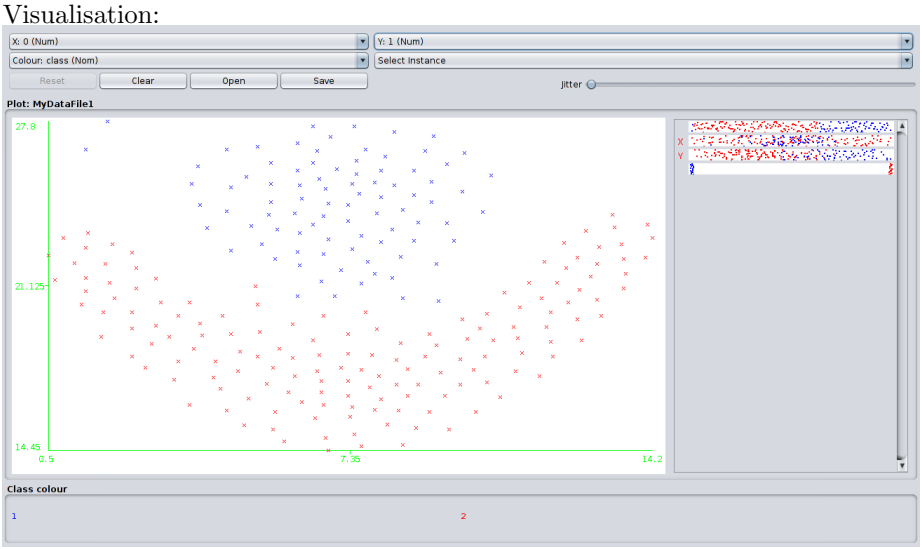
```
single link: Purity  100%
complete link:Purity  39%
average link:Purity 37 %
mean:Purity 40%
centroid: Purity 41 %
ward:Purity 41.5%
adjcomplete:Purity    36 %
neighbour join:Purity 34 %
best Singlelink:Purity 100%
```

Visualisation corresponding to best one:

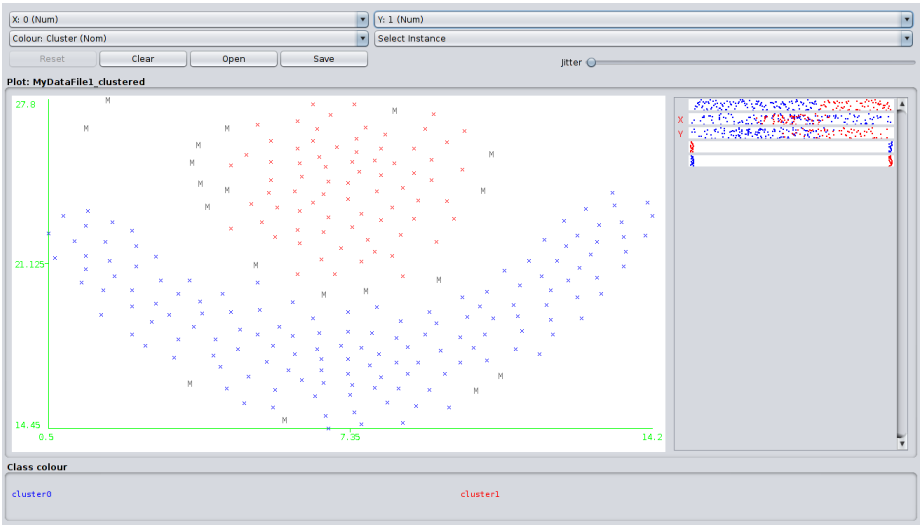


0.4 Flames

Visualisation:



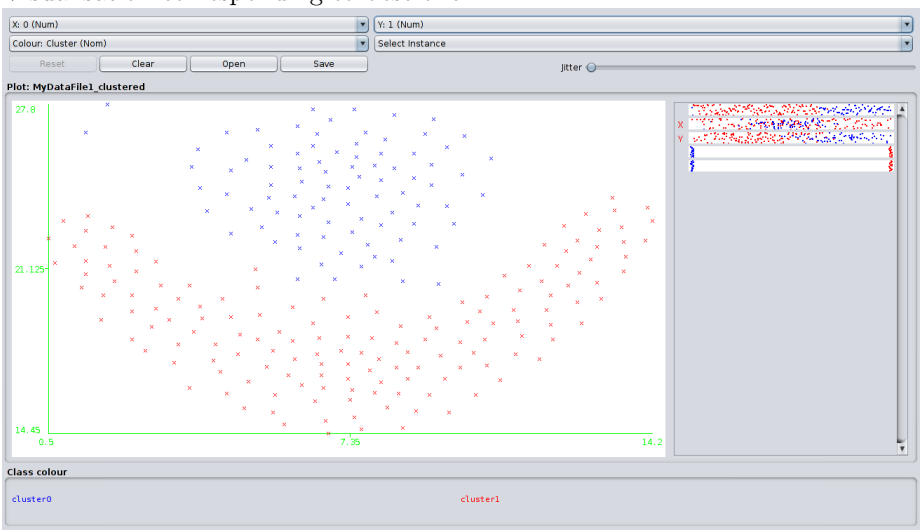
Best performance with DB Scan occured at:eps=0.07,m=6



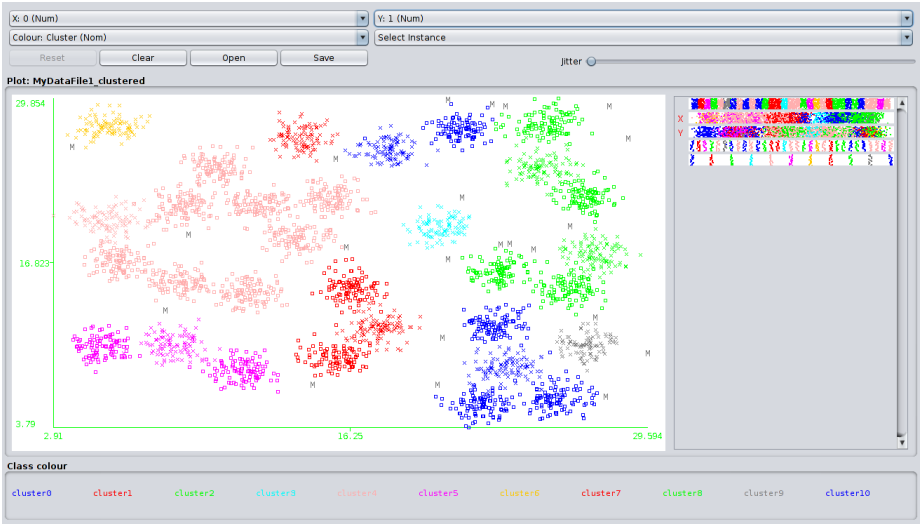
Incorrectly clustered instances : 0.0
Purity(*100)=100.00%)
Performances with following parameters for Hierarchial with Numclusters=3:

```
single link: Purity  65%
complete link:Purity  52%
average link:Purity 84 %
mean:Purity 92%
centroid: Purity 65 %
ward:Purity 100%
adjcomplete:Purity    65 %
neighbour join:Purity 63 %
best ward:Purity 100%
```

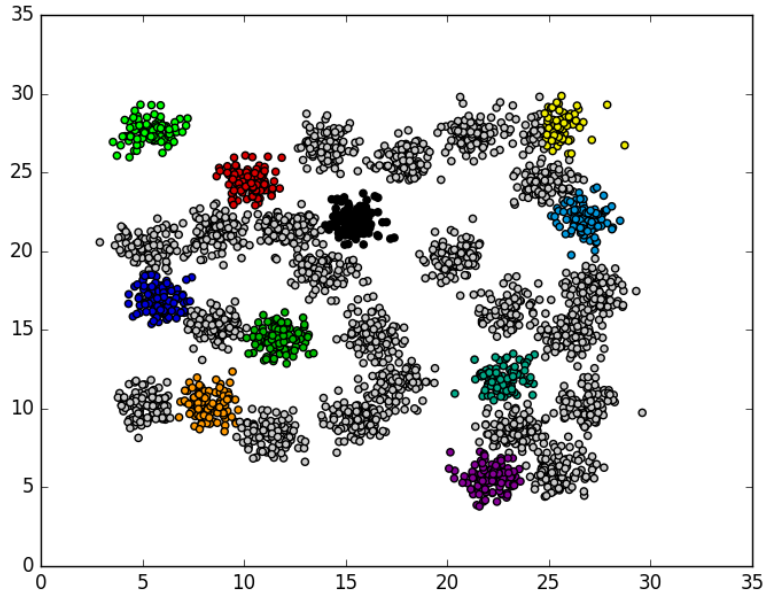
Visualisation corresponding to best one:



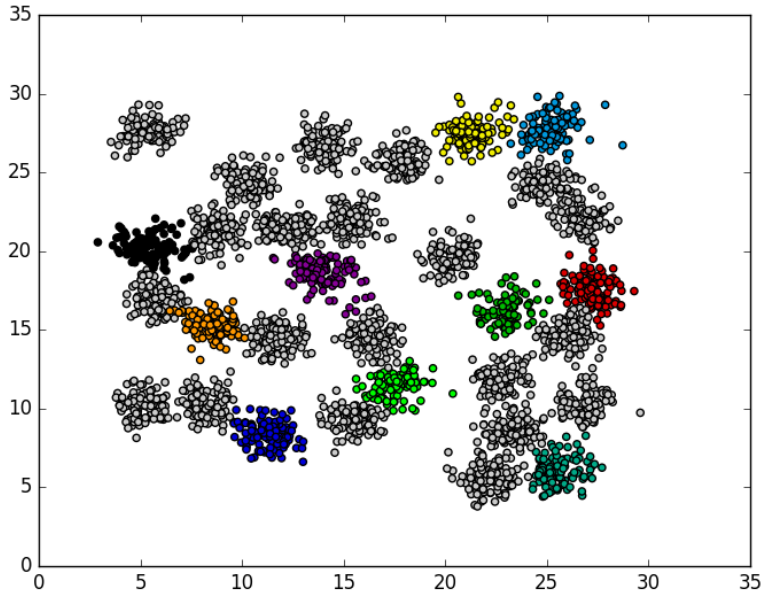
Now we will perform DBScan on D31 dataset(Best fit eps.=0.06,min.pts=6).



Purity(*100):56.36%
K-means with k=32 is not working in weka,so I did with sklearn.
Purity(*100)=96.5%
Still we didn't achieve 100% purity.Increasing k upto 40 also didnt help achieve 100%purity.



(Note:there is slight issue with colouring in mysklearn once clusters exceeds 12.Maybe 12 is max no of colours allowed) With ward,there is a slight increase in purity to96.9%



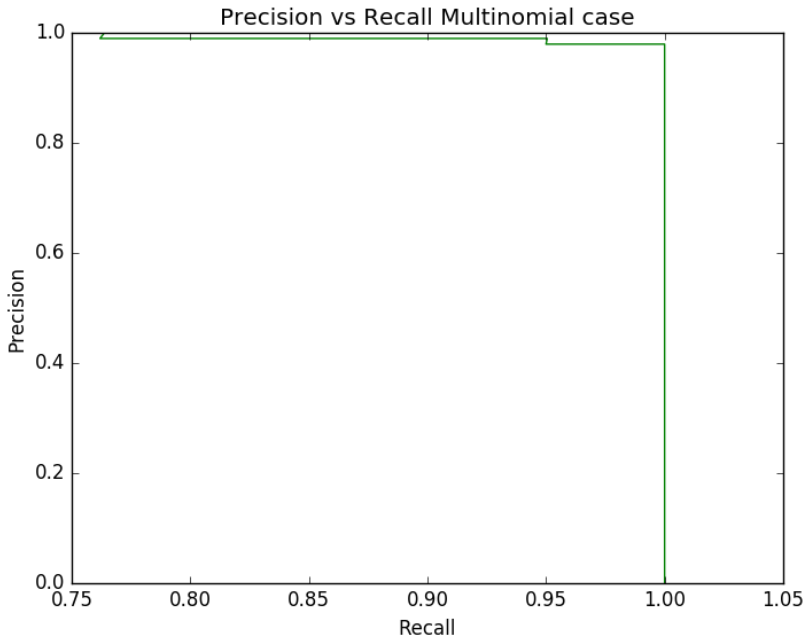
0.5 Question-2

Naive Bayes assumes independence between class conditioned probabilities of features.It is a reasonable assumption for text classification problem.In this problem we are assuming that position in which a particular word occurs doesn't make any difference.

0.5.1 A

Let C=spam class(the class of concern for us).
C=ham class
Now we are interested in P(C/x),where x is the input feature.We assume that x=[x_1,x_2...x_k]=[frequency of each word in a test mail y].
Now wkt P(C/x)=P(x/C)*P(C)/P(X). By multinomial Naive bayes assumption,P(x/c)=multinomial distribution(p_1,...p_k).
Now we have done in class,estimate(MLE) p_i=(No of occurances of word i in spam mail)/(total no of words in spam mails) ,evaluated using training data,ofcourse add ADD-ONE SMOOTHING.Now once we know these parameters,and P(C),we are done.
P(C)=(No of spam mails)/(total no of mails) [evaluated on training set]

We will then take log(before evaluating) as these values could go quite close to 0.
No compute P(C/X),assign it to C if it is greater than a threshold.

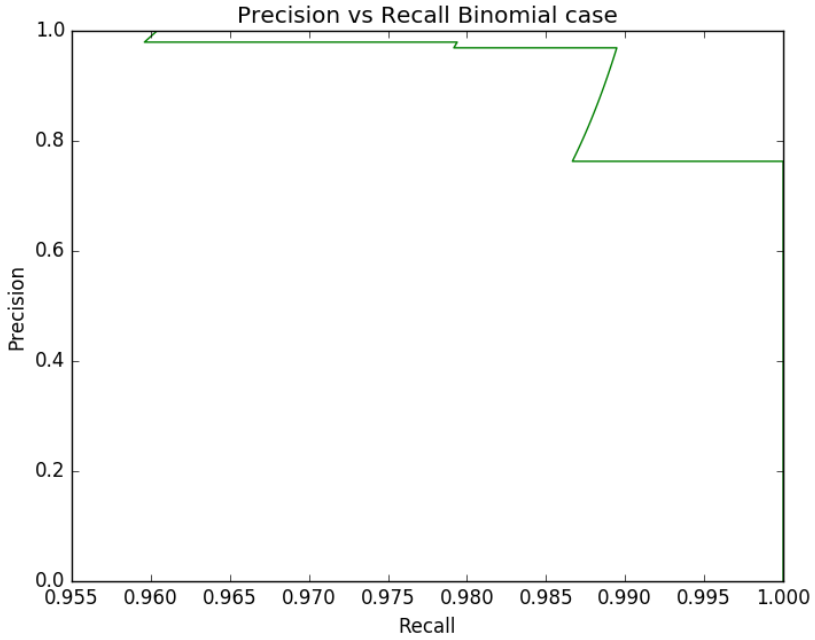


I implemented it from scratch.Following is Metrics:

```
 Class Precision   Recall   F-score   No.supports
  0.0       0.98      0.97      0.98        124
  1.0      0.96      0.98      0.97         97
-------------------------------------------
  0.0       0.99      0.96      0.98        124
  1.0       0.95      0.99      0.97         96
-------------------------------------------
  0.0       0.97      0.94      0.95         124
  1.0       0.92      0.96      0.94         96
-------------------------------------------
  0.0       0.97      0.96      0.96        123
  1.0       0.95      0.96      0.95         96
-------------------------------------------
  0.0       0.98      0.94      0.96        123
  1.0       0.93      0.98      0.95         96
-------------------------------------------
Avg Class1 Fscore:0.96
```

0.5.2 B

We do similar to part A,But perform binomial distribution,where each random variable is binomial distribution,with only 2 possible values,1-exists ,0-not exists.
P(C) will remail same in this case also,but P(x_i/C)=(no of spam mails in which word i)/(total no of spam mails) [evaluated with respect to training set].
All further derivations and assumptions are same.I also implemented this from scratch
The following is PR-Curve:



The following is metrics on testset:

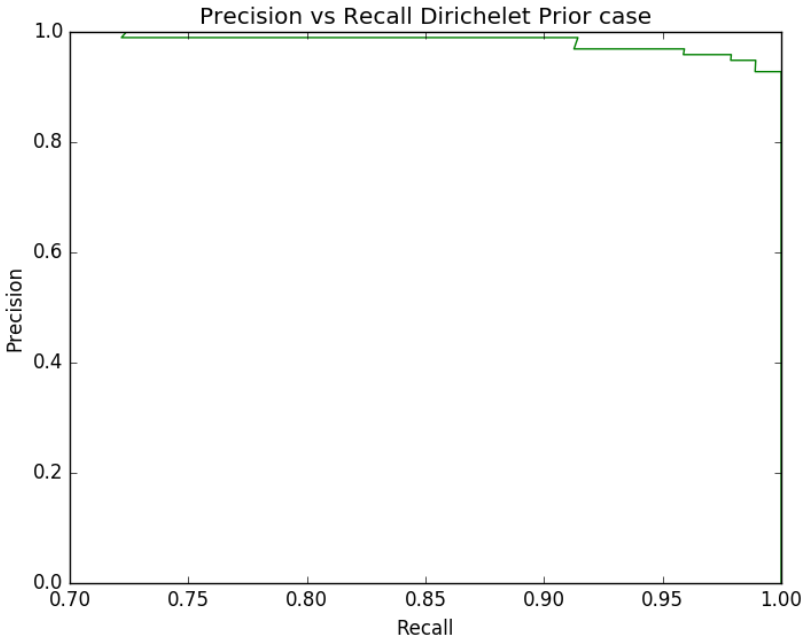
Class	Precision	Recall	F-score	No.supports
0.0	0.92	0.99	0.95	124
1.0	0.99	0.89	0.93	97
0.0	0.91	1.00	0.95	124
1.0	1.00	0.88	0.93	96

```
0.0         0.90        0.98        0.94         123
1.0         0.98        0.85        0.91          96
-------------------------------------------------
0.0         0.84        0.98        0.90         123
1.0         0.96        0.76        0.85          96
-------------------------------------------------
0.0         0.98        0.94        0.96         123
1.0         0.93        0.98        0.95          96
-------------------------------------------------
Avg Class1 Fscore:0.92
```

Obviously we have lost some information because of changing to bernoulli due to which we the avg F score dips.

0.6 C

We have estimated parameters,given beta distribution in class.Similarly in dirichlet distribution,we can show:
p_i=(number of occurances of word i in mail+α_1-1)/(total no of words in spam mails +$\sigma\alpha_i$-K).
But the constants go away due to add 1 smoothening.Prpeat same as part A
Heuristic 1:all α_i are uniform.One of the ways to validate this assumption is that we do not have the words,we only have their numbers,so applying this heuristic makes sense.
PR-Curve:



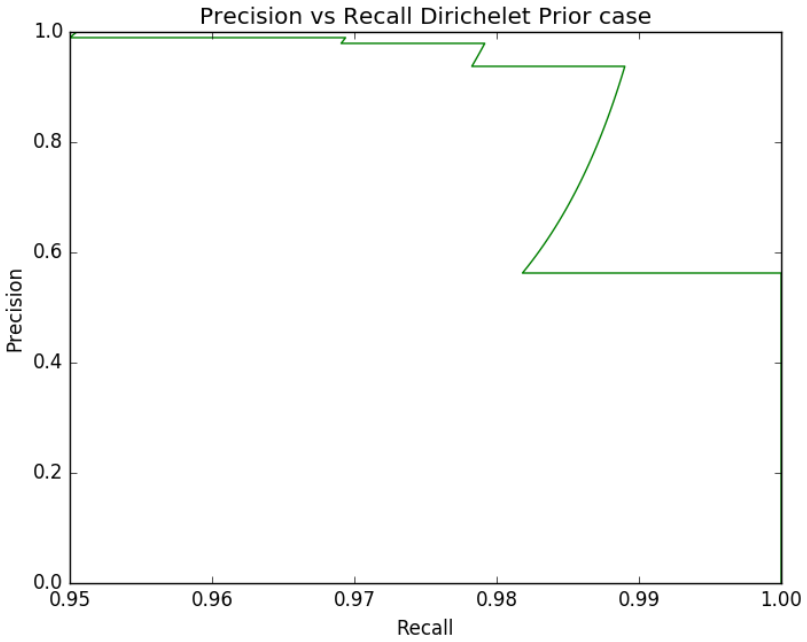
Class	Precision	Recall	F-score	No.supports
0.0	0.97	0.94	0.96	124
1.0	0.93	0.96	0.94	96

Class1 Fscore:0.94

So it has reduced,So try another heuristic:
I dont want to consider the words that appeared most frequent non spam class.I seems to make sense.Infact it is found to be best heuristic: α_i inversely proportional to freq in non-spam class
Applying this heuristic,we get following

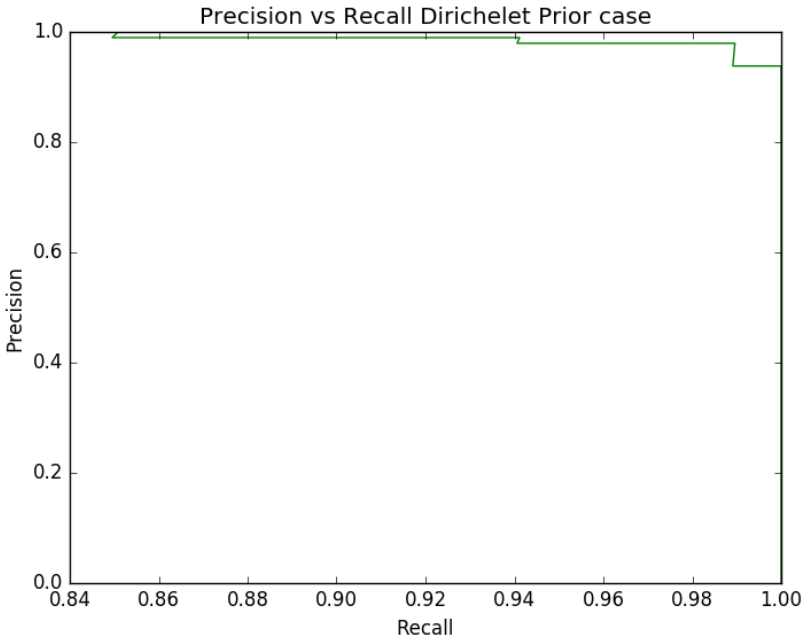
Class	Precision	Recall	F-score	No.supports
0.0	0.98	0.98	0.98	124
1.0	0.97	0.98	0.97	96

```
-------------------------------------------
Class1 Fscore:0.97
```



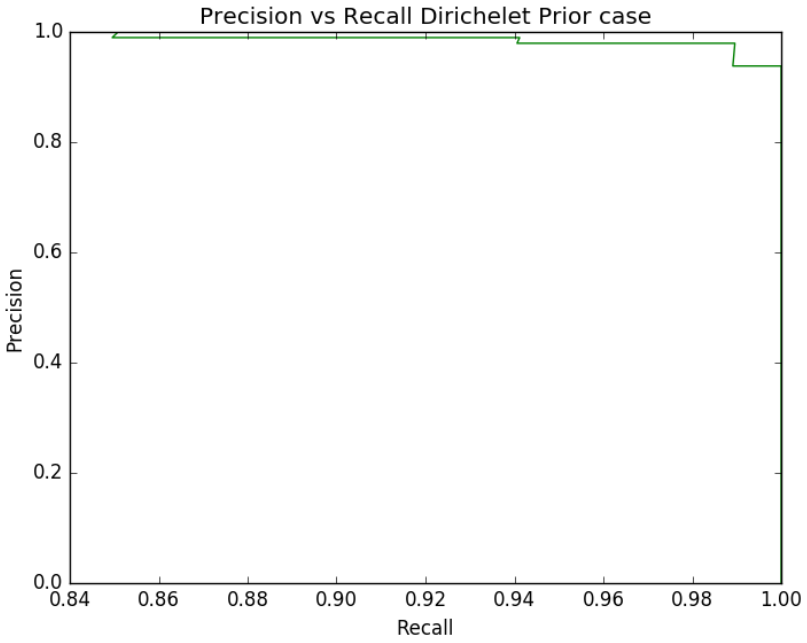
So it is an improvement.
C) what if i say a word is important if it occurs most in spam class.This also intutively makes sense
α_i proportional to freq in spam class

```
Class Precision   Recall   F-score   No.supports

0.0       0.98      0.96      0.97       124
1.0       0.95      0.98      0.96        97

-------------------------------------------
Class1 Fscore:0.96
```



Which is not much of improvement.
D) what if i combine B),C) heuristics?with some proportionality parameter. It seems that there isnt much change.

```
Class Precision   Recall   F-score   No.supports

0.0       0.99      0.96      0.98       124
1.0       0.95      0.99      0.97        96
-----------------------------------------
Class1 Fscore:0.97
```



0.6.1 D

Assume class prior follow beta distribution.We already solved in class,p_i=(n1+α-1)/(n1+n2+α+β-2)
Now we have no domain knowledge,So perform grid search to get optimal α,β.
vary alpha=[100,200..2000],beta=[100,200,2000]
Following is result obtained(a is alpha,b is beta):

```
a=2000,b=1000
precision    recall  f1-score   support

0.0       0.98      0.97      0.98       124
1.0       0.96      0.98      0.97        96

avg / total       0.97      0.97      0.97       220
a=2000,b=500
precision    recall  f1-score   support

0.0       0.97      0.94      0.95       124
1.0       0.92      0.96      0.94        96

avg / total       0.95      0.95      0.95       220
a=2000,b=200
```

precision	recall	f1-score	support
0.0	0.98	0.94	0.96	123
1.0	0.93	0.98	0.95	96
avg / total	0.96	0.96	0.96	219

α=2000,β=1000 seems to be the best possible solutions.
Following is PR curve for α=2000,β=1000: