Procedure

Dataset:

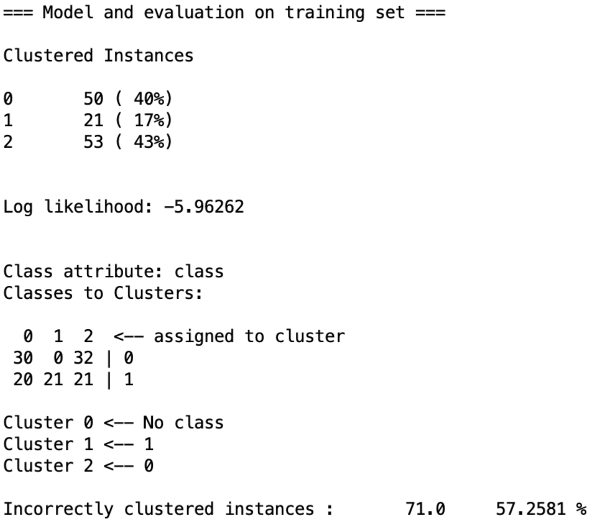
This is an artificial dataset with 124 instances, each described by 6 discrete attributes and a binary class attribute.
There are 3 versions of the so-called monk problem: We are using the first version (MONK-1: (a1 = a2) or (a5 = 1)) in this exercise.

Clusters may not correspond to classes:

First, cluster the data with different algorithms and number of clusters. Use the Clusters to class evaluation model to see whether the clustering algorithm is able to discover the class division existing in the data.

By applying EM algorithm and Simple K-means algorithm with 3 clusters and 5 clusters to discover the class division. And we also applied the quality of classification. The results are shown as below.

EM algorithm with 3 clusters



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

Clustered Instances

0       50 ( 40%)
1       21 ( 17%)
2       53 ( 43%)


Log likelihood: -5.96262


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
 30  0 32 | 0
 20 21 21 | 1

Cluster 0 <-- No class
Cluster 1 <-- 1
Cluster 2 <-- 0

Incorrectly clustered instances :       71.0     57.2581 %
```

EM algorithm with 5 clusters

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

Clustered Instances

0       41 ( 33%)
1       31 ( 25%)
2       21 ( 17%)
3       17 ( 14%)
4       14 ( 11%)


Log likelihood: -5.91866


Class attribute: class
Classes to Clusters:

  0  1  2  3  4  <-- assigned to cluster
 21 26  4  0 11 | 0
 20  5 17 17  3 | 1

Cluster 0 <-- 1
Cluster 1 <-- 0
Cluster 2 <-- No class
Cluster 3 <-- No class
Cluster 4 <-- No class

Incorrectly clustered instances :       78.0     62.9032 %
```

Simple K-means algorithm with 3 clusters

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

Clustered Instances

0       52 ( 42%)
1       44 ( 35%)
2       28 ( 23%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
 25 27 10 | 0
 27 17 18 | 1

Cluster 0 <-- 1
Cluster 1 <-- 0
Cluster 2 <-- No class

Incorrectly clustered instances :       70.0     56.4516 %
```

Simple K-means algorithm with 5 clusters

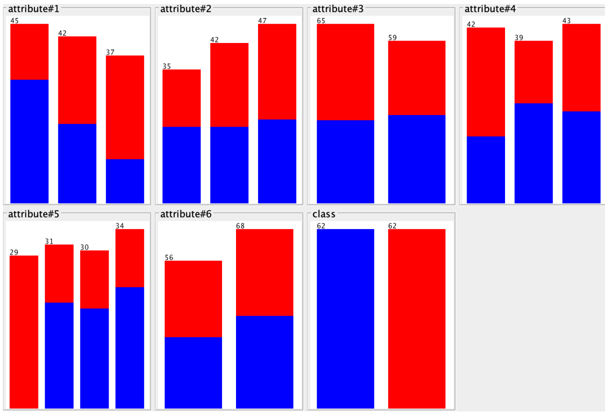
```
=== Model and evaluation on training set ===

Clustered Instances

0       38 ( 31%)
1       31 ( 25%)
2       19 ( 15%)
3       17 ( 14%)
4       19 ( 15%)


Class attribute: class
Classes to Clusters:

  0  1  2  3  4  <-- assigned to cluster
 20 21  2 12  7 | 0
 18 10 17  5 12 | 1

Cluster 0 <-- 1
Cluster 1 <-- 0
Cluster 2 <-- No class
Cluster 3 <-- No class
Cluster 4 <-- No class

Incorrectly clustered instances :       85.0     68.5484 %
```

From the result we can notice that the percentage of incorrectly clustered instances are very high among all situations. All the situation have error rate over 50 percent. And when the clustering algorithm is the same, the percentage of incorrectly clustered instances will go up with the increases of number of clusters.

So we can conclude the clustering algorithm is not enough to discover the existing class division.

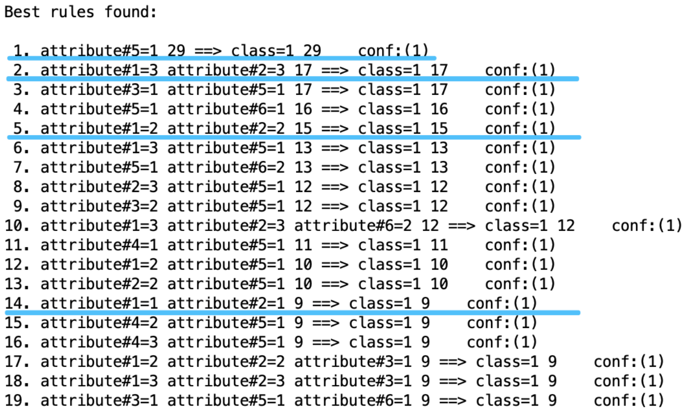
Why can the clustering algorithms not find a clustering that matches the class division in the database?

As we can see form the summary of data as following figure, the percentage of each class are almost the same in each attribute. So it will be hard to find the sign of the clusters to match the classes in this data set.



Use association analysis to find a set of rules that are able to accurately predict the class label from the rest of the attributes. Try to find as few rules predicting class 1 as possible.

As the figure shows below, No. 1, 2, 4, 14 are the best rules which meet the first vision of monk problems: MONK-1: (a1 = a2) or (a5 = 1).



```
Best rules found:

 1. attribute#5=1 29 ==> class=1 29    conf:(1)
 2. attribute#1=3 attribute#2=3 17 ==> class=1 17    conf:(1)
 3. attribute#3=1 attribute#5=1 17 ==> class=1 17    conf:(1)
 4. attribute#5=1 attribute#6=1 16 ==> class=1 16    conf:(1)
 5. attribute#1=2 attribute#2=2 15 ==> class=1 15    conf:(1)
 6. attribute#1=3 attribute#5=1 13 ==> class=1 13    conf:(1)
 7. attribute#5=1 attribute#6=2 13 ==> class=1 13    conf:(1)
 8. attribute#2=3 attribute#5=1 12 ==> class=1 12    conf:(1)
 9. attribute#3=2 attribute#5=1 12 ==> class=1 12    conf:(1)
10. attribute#1=3 attribute#2=3 attribute#6=2 12 ==> class=1 12    conf:(1)
11. attribute#4=1 attribute#5=1 11 ==> class=1 11    conf:(1)
12. attribute#1=2 attribute#5=1 10 ==> class=1 10    conf:(1)
13. attribute#2=2 attribute#5=1 10 ==> class=1 10    conf:(1)
14. attribute#1=1 attribute#2=1 9 ==> class=1 9    conf:(1)
15. attribute#4=2 attribute#5=1 9 ==> class=1 9    conf:(1)
16. attribute#4=3 attribute#5=1 9 ==> class=1 9    conf:(1)
17. attribute#1=2 attribute#2=2 attribute#3=1 9 ==> class=1 9    conf:(1)
18. attribute#1=3 attribute#2=3 attribute#3=1 9 ==> class=1 9    conf:(1)
19. attribute#3=1 attribute#5=1 attribute#6=1 9 ==> class=1 9    conf:(1)
```

Finally, would you say that the clustering algorithms fail or perform poorly for the monk1 dataset? Why or why not?

From the analysis above, we have to say the clustering algorithms failed for the monk1 dataset. Because the class in data here is binary, however, the algorithms use the distance method to classified the data. So it will be hard for algorithms to work on binary dataset here.