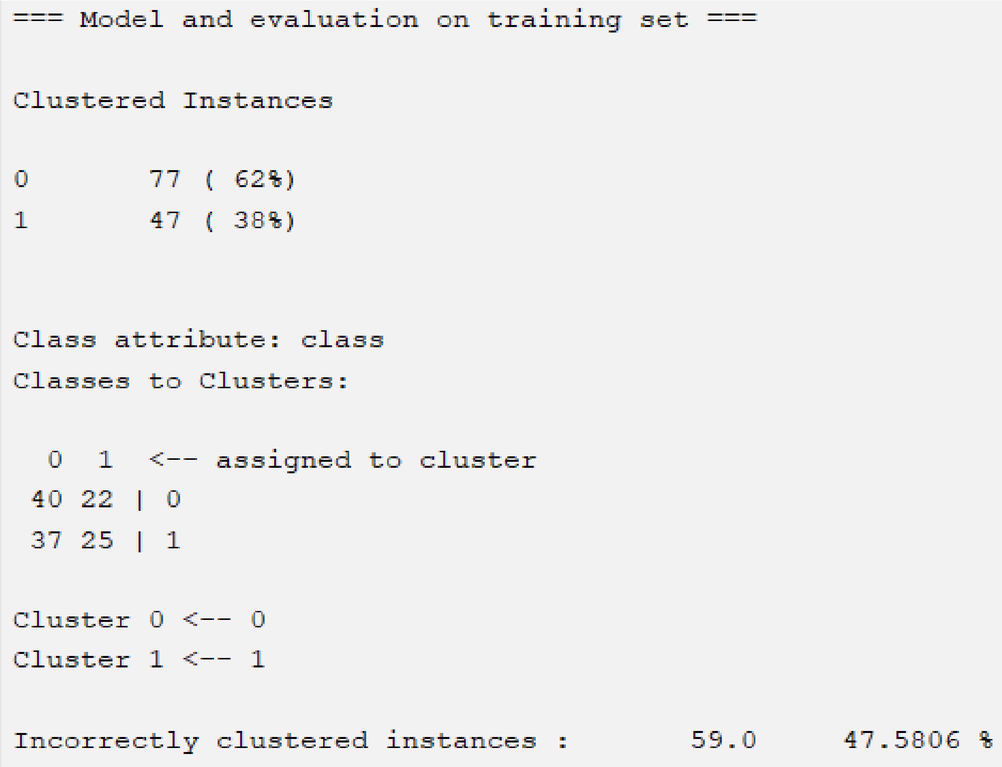
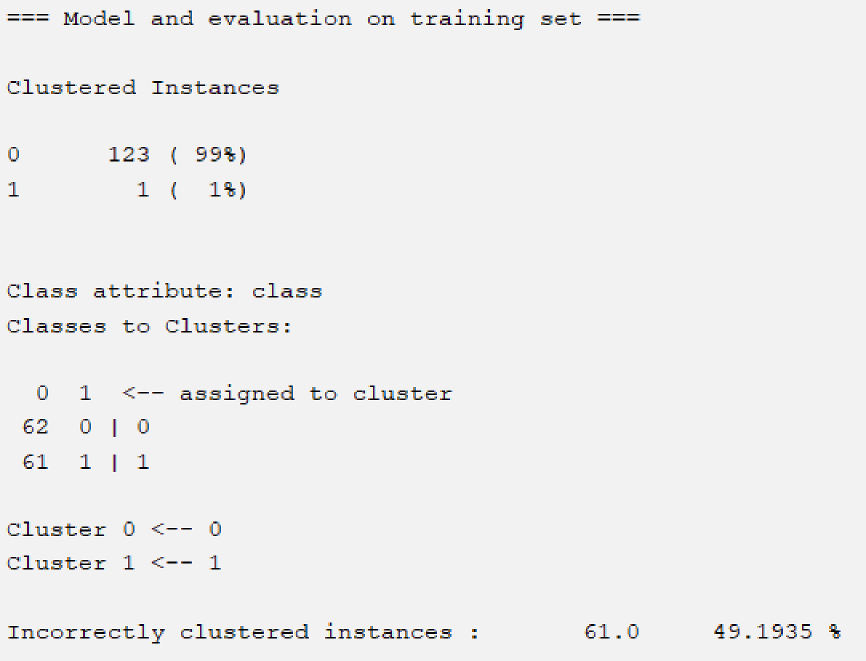
Lab 3 Group 17

Cluster the data with different algorithms and number of clusters**.** In this part, we chose SimpleKmean and Hierarchical Clustering with all default settings and experimented with 2 clusters first and 3 clusters.



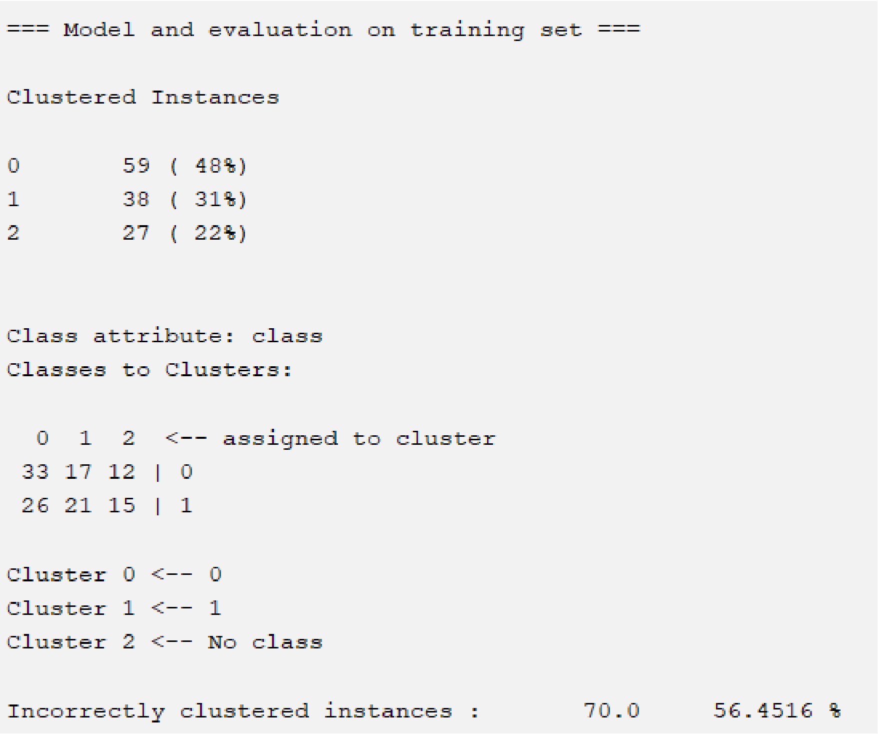
```
=== Model and evaluation on training set ===

Clustered Instances

0       77 ( 62%)
1       47 ( 38%)


Class attribute: class
Classes to Clusters:

  0  1  <-- assigned to cluster
 40 22 | 0
 37 25 | 1

Cluster 0 <-- 0
Cluster 1 <-- 1

Incorrectly clustered instances :        59.0     47.5806 %
```

SimpleKmean with 2 clusters



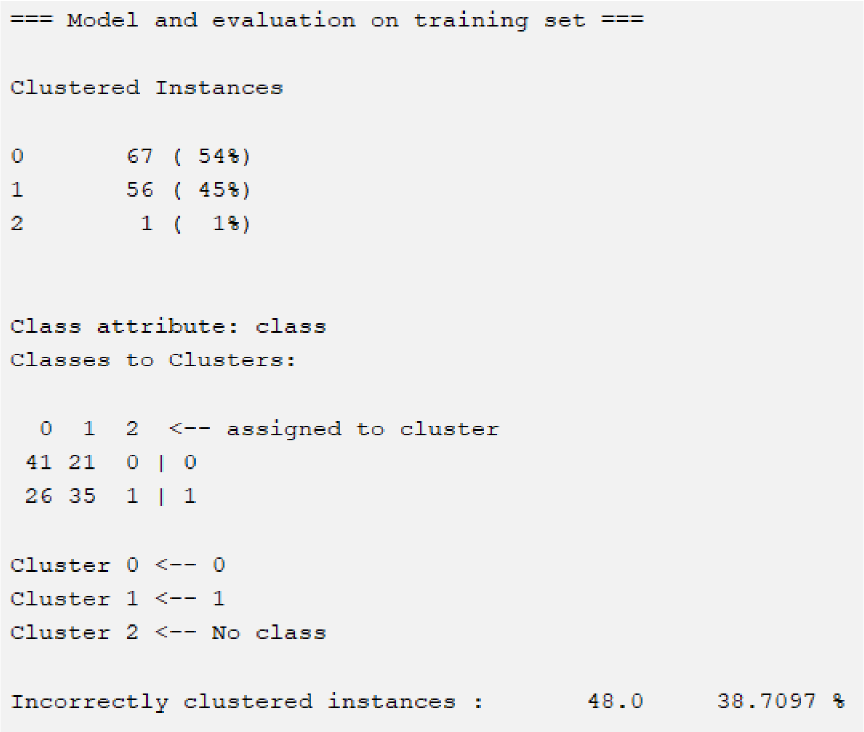
```
=== Model and evaluation on training set ===

Clustered Instances

0      123 ( 99%)
1        1 (  1%)


Class attribute: class
Classes to Clusters:

  0  1  <-- assigned to cluster
 62  0 | 0
 61  1 | 1

Cluster 0 <-- 0
Cluster 1 <-- 1

Incorrectly clustered instances :       61.0     49.1935 %
```

Hierarchical Clustering with 2 clusters



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

Clustered Instances

0       59 ( 48%)
1       38 ( 31%)
2       27 ( 22%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
 33 17 12 | 0
 26 21 15 | 1

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

Incorrectly clustered instances :	70.0	 56.4516 %
```

SimpleKmean with 3 clusters



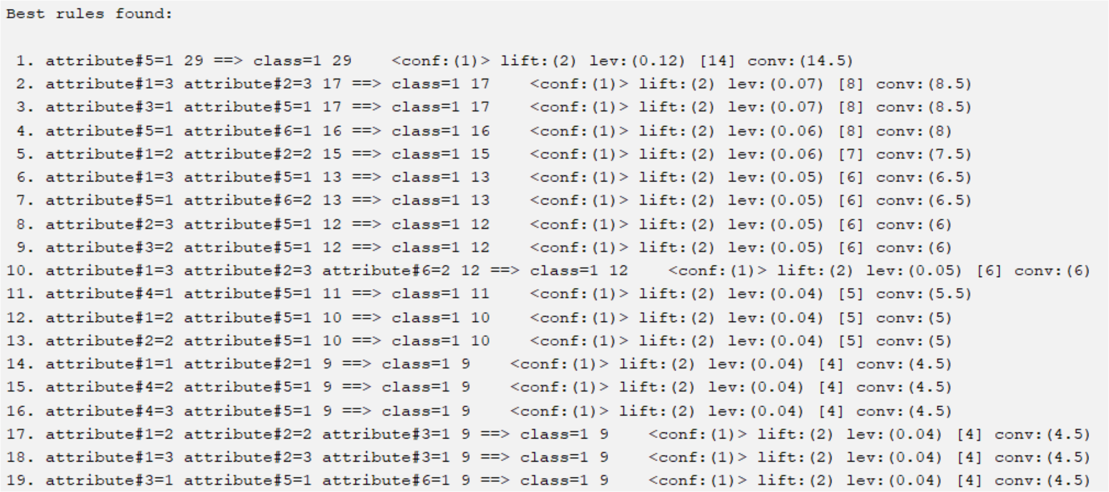
```
=== Model and evaluation on training set ===

Clustered Instances

0       67 ( 54%)
1       56 ( 45%)
2        1 (  1%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
 41 21  0 | 0
 26 35  1 | 1

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

Incorrectly clustered instances :	48.0	 38.7097 %
```

Hierarchical Clustering with 3 clusters

By observing the result from 2 clusters, both clustering algorithms do not find the class division in the data. Both algorithms result in around 50% accuracy.
Then we have tried to add one more cluster to see what would happen. We observed

that in 3 clusters Hierarchical Clustering the result got some improvement. This might be the algorithm separating some outliers into new clusters. However, overall the clustering algorithms did not achieve ideal results. The clustering algorithm does not get good result because the algorithm only takes the similarity of data points into account but doesn't take their label into account.

We then use association analysis to find a set of rules that are able to accurately predict the class label from the rest of the attributes. We use a minimum support of 0.05 and a maximum number of rules of 19, as recommended.



```
Best rules found:

 1. attribute#5=1 29 ==> class=1 29    <conf:(1)> lift:(2) lev:(0.12) [14] conv:(14.5)
 2. attribute#1=3 attribute#2=3 17 ==> class=1 17    <conf:(1)> lift:(2) lev:(0.07) [8] conv:(8.5)
 3. attribute#3=1 attribute#5=1 17 ==> class=1 17    <conf:(1)> lift:(2) lev:(0.07) [8] conv:(8.5)
 4. attribute#5=1 attribute#6=1 16 ==> class=1 16    <conf:(1)> lift:(2) lev:(0.06) [8] conv:(8)
 5. attribute#1=2 attribute#2=2 15 ==> class=1 15    <conf:(1)> lift:(2) lev:(0.06) [7] conv:(7.5)
 6. attribute#1=3 attribute#5=1 13 ==> class=1 13    <conf:(1)> lift:(2) lev:(0.05) [6] conv:(6.5)
 7. attribute#5=1 attribute#6=2 13 ==> class=1 13    <conf:(1)> lift:(2) lev:(0.05) [6] conv:(6.5)
 8. attribute#2=3 attribute#5=1 12 ==> class=1 12    <conf:(1)> lift:(2) lev:(0.05) [6] conv:(6)
 9. attribute#3=2 attribute#5=1 12 ==> class=1 12    <conf:(1)> lift:(2) lev:(0.05) [6] conv:(6)
10. attribute#1=3 attribute#2=3 attribute#6=2 12 ==> class=1 12    <conf:(1)> lift:(2) lev:(0.05) [6] conv:(6)
11. attribute#4=1 attribute#5=1 11 ==> class=1 11    <conf:(1)> lift:(2) lev:(0.04) [5] conv:(5.5)
12. attribute#1=2 attribute#5=1 10 ==> class=1 10    <conf:(1)> lift:(2) lev:(0.04) [5] conv:(5)
13. attribute#2=2 attribute#5=1 10 ==> class=1 10    <conf:(1)> lift:(2) lev:(0.04) [5] conv:(5)
14. attribute#1=1 attribute#2=1 9 ==> class=1 9    <conf:(1)> lift:(2) lev:(0.04) [4] conv:(4.5)
15. attribute#4=2 attribute#5=1 9 ==> class=1 9    <conf:(1)> lift:(2) lev:(0.04) [4] conv:(4.5)
16. attribute#4=3 attribute#5=1 9 ==> class=1 9    <conf:(1)> lift:(2) lev:(0.04) [4] conv:(4.5)
17. attribute#1=2 attribute#2=2 attribute#3=1 9 ==> class=1 9    <conf:(1)> lift:(2) lev:(0.04) [4] conv:(4.5)
18. attribute#1=3 attribute#2=3 attribute#3=1 9 ==> class=1 9    <conf:(1)> lift:(2) lev:(0.04) [4] conv:(4.5)
19. attribute#3=1 attribute#5=1 attribute#6=1 9 ==> class=1 9    <conf:(1)> lift:(2) lev:(0.04) [4] conv:(4.5)
```

Above is the best 19 rules apriori algorithm find. Since we need to find as few rules that can be used to classify classes 0 and 1, we look at the data set and find out which four rules can cover all class 1.

All the best rules here are fully supported, i.e., if the rule is satisfied, it will classify as 1. Therefore, we work from the top and mark the data that fit the rules.

After rules 1 and 2, we see the data that fits rules 3 and 4 is already marked since the first rule already covers all attribute#5 = 1. Therefore we move on to rule 5 and can easily see after rule 5, the rest of the data fits rule 14.

In the end, we find out rules **1,2,5,14** can cover all the class 1.

We consider clustering algorithm failed for the monk 1 data set. As the reason we gave above, the clustering algorithm will not consider the existing label of the data point. Moreover, some attributes here do not directly relate to the label. As we can see from the association analysis, we have to combine different attributes as a rule to classify the classes. Since the clustering algorithm is only comparing the similarity of each attribute, this is the reason why the clustering algorithm failed