TDDD41-732A75 Data Mining - Clustering and Association Analysis and Advanced Data Mining

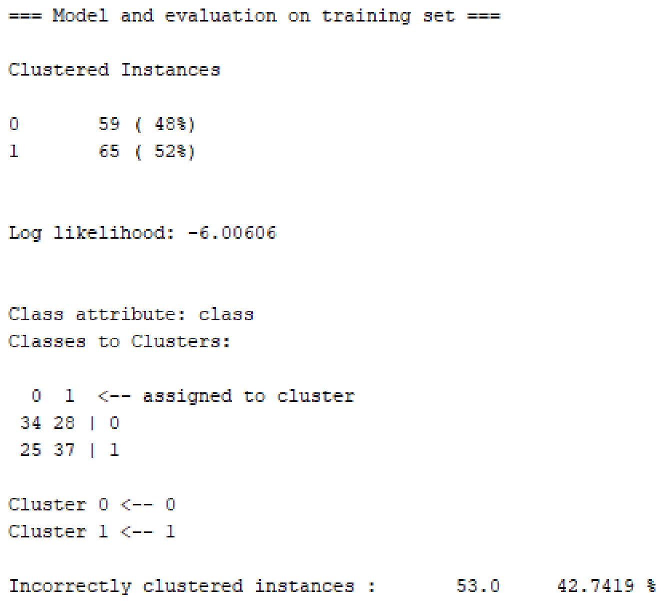
Pranav Pankaj Chandode (prach896)

Varun Gurupurandar (vargu125)

Lab 3 Association Analysis 2

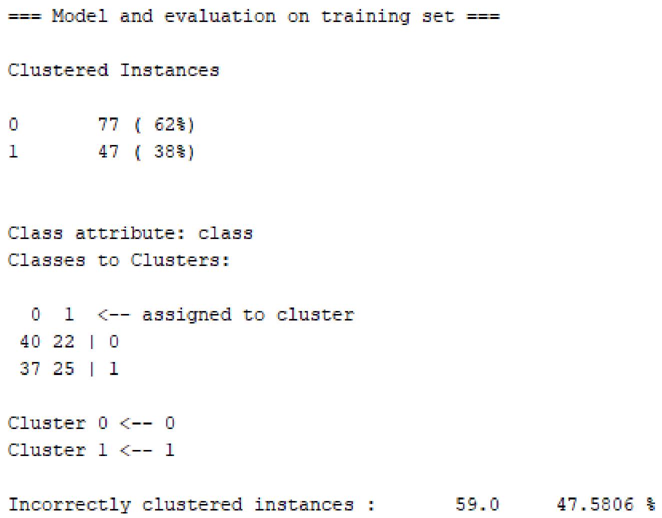
Clustering on Monk1 dataset

We were tasked with using different clustering algorithms and evaluating them against the class labels to determine if a simple clustering algorithm can capture the underlying class divisions. To do this, we used the Expectation-Maximization (EM) algorithm and the KMeans algorithm. Our first clustering attempt with EM and two clusters resulted in a classification error of 42.7%.



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

Clustered Instances

0       59 ( 48%)
1       65 ( 52%)


Log likelihood: -6.00606


Class attribute: class
Classes to Clusters:

  0  1  <-- assigned to cluster
 34 28 | 0
 25 37 | 1

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

Incorrectly clustered instances :       53.0     42.7419 %
```

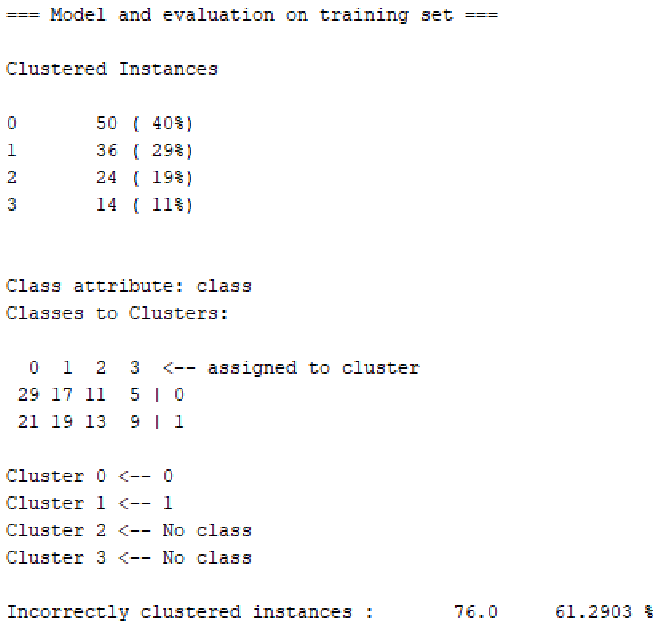
For the second attempt we used KMeans with k = 2



```
=== 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 %
```

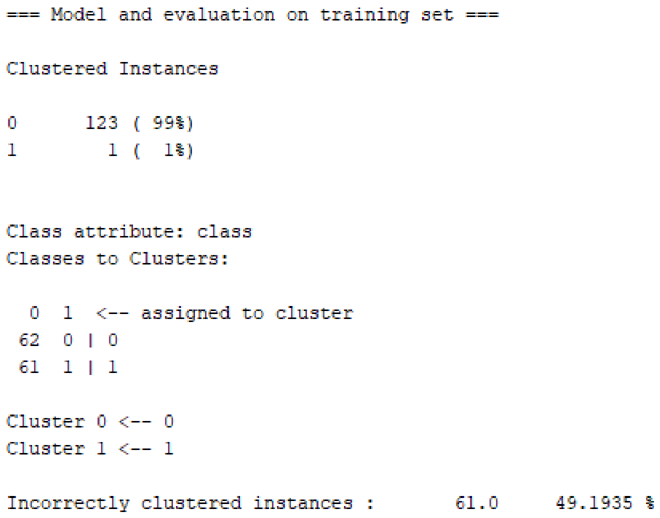
The results were worse than the EM algorithm with 47.58% classification error.

Following the exercise instructions, we also tried clustering with more than two clusters. However, since our target variable is binary, evaluating the results became more difficult. The classification error considers any instances not assigned to cluster 0 or 1 as misclassified. When we looked at clusters 2 and 3, we found that there was no clear distinction between the classes, as at least one-third of the data points in each cluster belonged to the minority class. As a result, we can conclude that this approach also failed to capture a meaningful class separation.



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

Clustered Instances

0       50 ( 40%)
1       36 ( 29%)
2       24 ( 19%)
3       14 ( 11%)


Class attribute: class
Classes to Clusters:

  0  1  2  3  <-- assigned to cluster
 29 17 11  5 | 0
 21 19 13  9 | 1

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

Incorrectly clustered instances :       76.0     61.2903 %
```

Finally, we implemented hierarchical clustering with 2 clusters, and got the following result;



```
=== 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 %
```

This case gave a classification error as good as a naïve classifier

Association Analysis

We created rules using the apriori algorithm with a support value of 0.05 and confidence 1 and maximum of 19 rules. Among the 19 rules, 4 rules described the class 1 completely. The rules were as follows:

1. attribute#5=1 29 ==⇒ class=1 29 conf:(1)

2. attribute#1=1 attribute#2=1 9 ==⇒ class=1 9 conf:(1)

3. attribute#1=2 attribute#2=2 15 ==⇒ class=1 15 conf:(1)

4. attribute#1=3 attribute#2=3 17 ==⇒ class=1 17 conf:(1)

Questions

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

➔ The consistent mixing of classes across all configurations arises from a fundamental mismatch between the clustering algorithms' approaches and the dataset's structure. SimpleKMeans groups instances based on the Euclidean distance metric, which measures numerical similarity between attribute values but cannot identify logical relationships, such as attribute equality. EM, which takes a probabilistic approach based on attribute distributions, performs slightly better but still fails to recognize conditional patterns.
When analyzing our rules, the first rule is based on a single attribute value. The other three rules are composite, combining attribute#1 and attribute#2. These can be simplified to attribute#1 = attribute#2. If we were to plot all the data points not captured by the first rule in a two-dimensional space defined

by attribute#1 and attribute#2, the data points belonging to class 1 would align along the diagonal.
This implies that data points from class 1 with the value pair (1, 1) would be "closer" to class 0 data points than to class 1 data points with the value pair (3, 3). This demonstrates that distance is not the key factor in this case.
Furthermore, attributes #3, #4, and #6 have no influence on class boundaries, yet they would still affect every clustering algorithm because they influence the concept of distance or proximity.

2. Would you say that the clustering algorithms fail or perform poorly for the monk1 dataset? Why or why not?

➔ The poor performance can be attributed to the design of the dataset, which incorporates a logical pattern that doesn't align with the metrics used by SimpleKMeans, hierarchical and EM. As shown by the association analysis, the classes in the monk1 dataset are governed by conditional rules rather than spatial or distributional clustering. As a result, these algorithms fail to achieve the exercise's goal of matching the predefined class structure, as they are not suited to interpret the logical relationships that define the data. This outcome highlights the importance of selecting analytical methods that are in line with the inherent characteristics of the dataset, a point further emphasized by the success of the Apriori algorithm in capturing the exact rules governing class 1.