Module Code: CS3DS19

Student Number: 28017133
Date (when the work completed): 29/03/2022
Actual hrs spent for the assignment: 13

Introduction

In this project, a clustering analysis is performed for the multidimensional ‘wine’ data set. The dataset (wine.csv) is based on a chemical examination of three distinct varieties of wine cultivated in the same region of Italy. The amount of 13 chemical compounds detected in each wine was established during the study. The cultivar ID (1, 2 or 3) and 13 numerical characteristics are included in each data record. The analysis is performed in two tasks. For the first task, clustering is performed on the original data and for the second task the data is normalized.

Task 1: Clustering without normalization

The KNIME workflow for Task 1 is shown in Figure 1. First, the dataset containing the information about the wines is read through the CSV reader node. The Color Manager node is used to apply color to the different wine classes to make them visible. Principal Component Analysis (PCA) is used to reduce the dimensions of the data. PCA aims to find the directions of maximum variance in high-dimensional data and projects it onto a new subspace with fewer dimensions than the original one. The PCA node is configured accordingly to reduce the dataset to two dimensions so that the data can be viewed on a 2D Plot. The Column Resorter node is used next in order to move the two new principal component dimensions first so that they will be plotted by default on the scatter plot, which is then plotted using the Scatter Plot node. The Scatter Plot produced is shown in Figure 2.

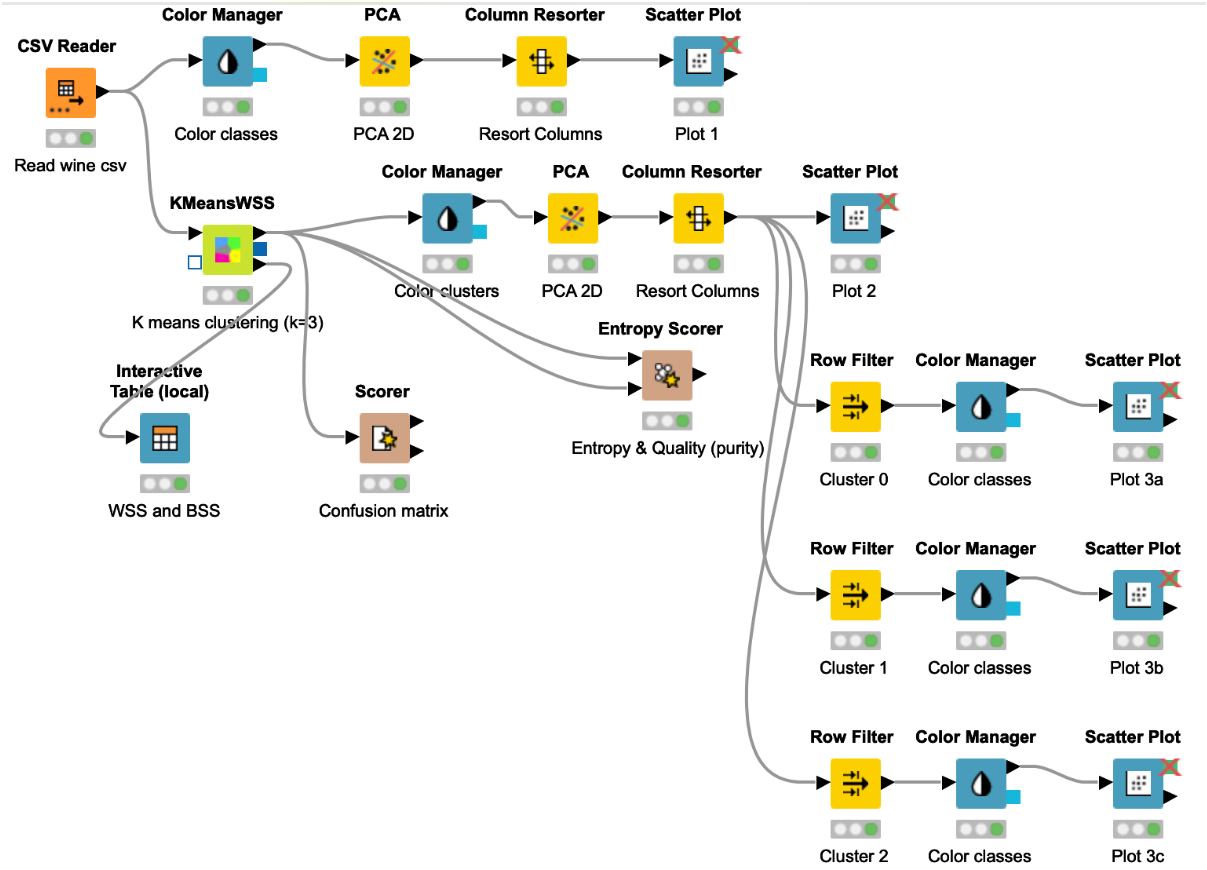


Figure 1: Task 1 Workflow

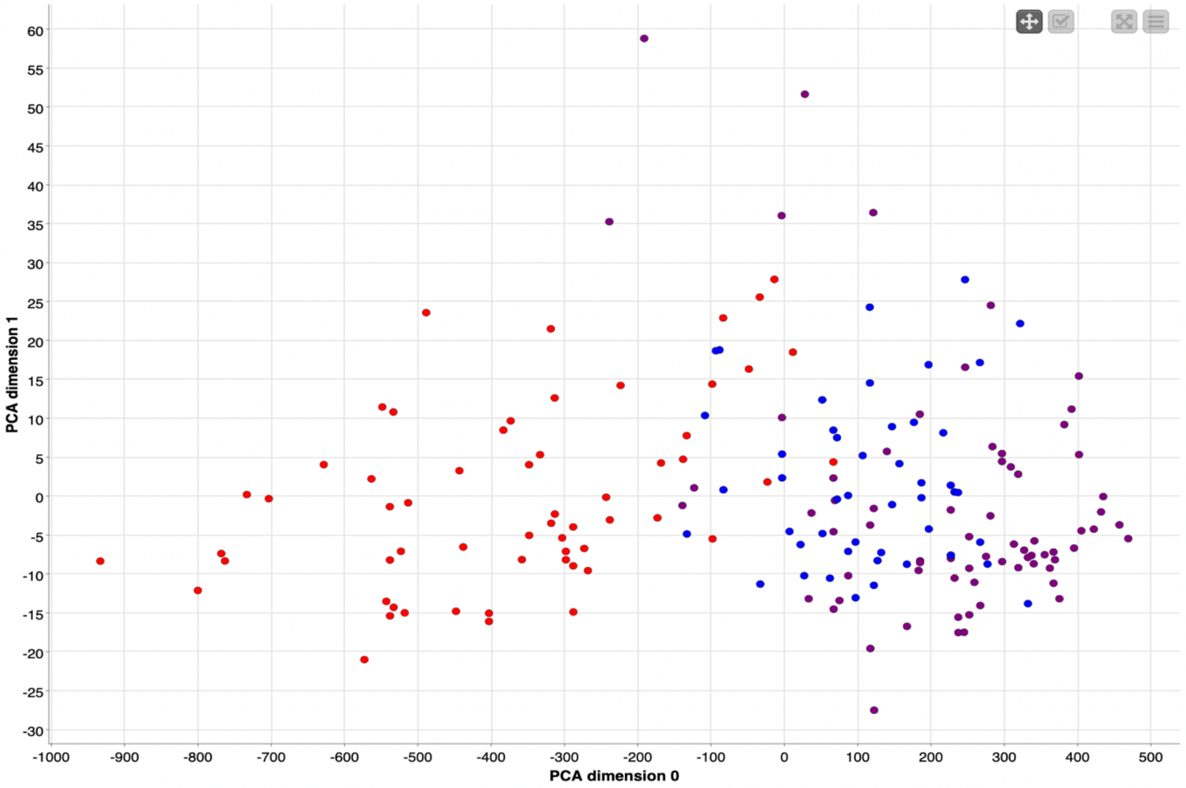


Figure 2: Scatter Plot of Classes (task 1, plot 1)

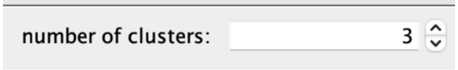
The Scatter Plot in Figure 2 shows the three classes of wine as follows:

Red: class 1
Purple: class 2
Blue: class 3

The clustering algorithm adopted is the k-means clustering. It is an unsupervised learning algorithm which groups the dataset into different clusters. K defines the number of pre-defined clusters that need to be created in the process. In our case, we wish to generate three partitions so k will equal to 3. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. The steps followed by the k-means algorithm are (Javapoint, 2021):

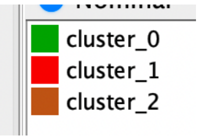
1. Select the number K to decide the number of clusters.
2. Select random K points or centroids.
3. Assign each data point to their closest centroid, which will form the predefined K clusters.
4. Calculate the variance and place a new centroid of each cluster.
5. Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.
6. If any reassignment occurs, then go to step-4 else go to FINISH.

The implementation of the k-means clustering is done using a slightly modified version of the K-Means algorithm node, provided by the University of Reading. The configuration of the node is as follows:



number of clusters: 3

After the clustering was performed, using the Color Manager node the three cluster were assigned colors.



🟩 cluster_0
🟥 cluster_1
🟫 cluster_2

As before, Principal Component Analysis was implemented to plot the three clusters onto a two-dimensional space. The columns were resorted, and the Scatter Plot produced is shown in Figure 3.

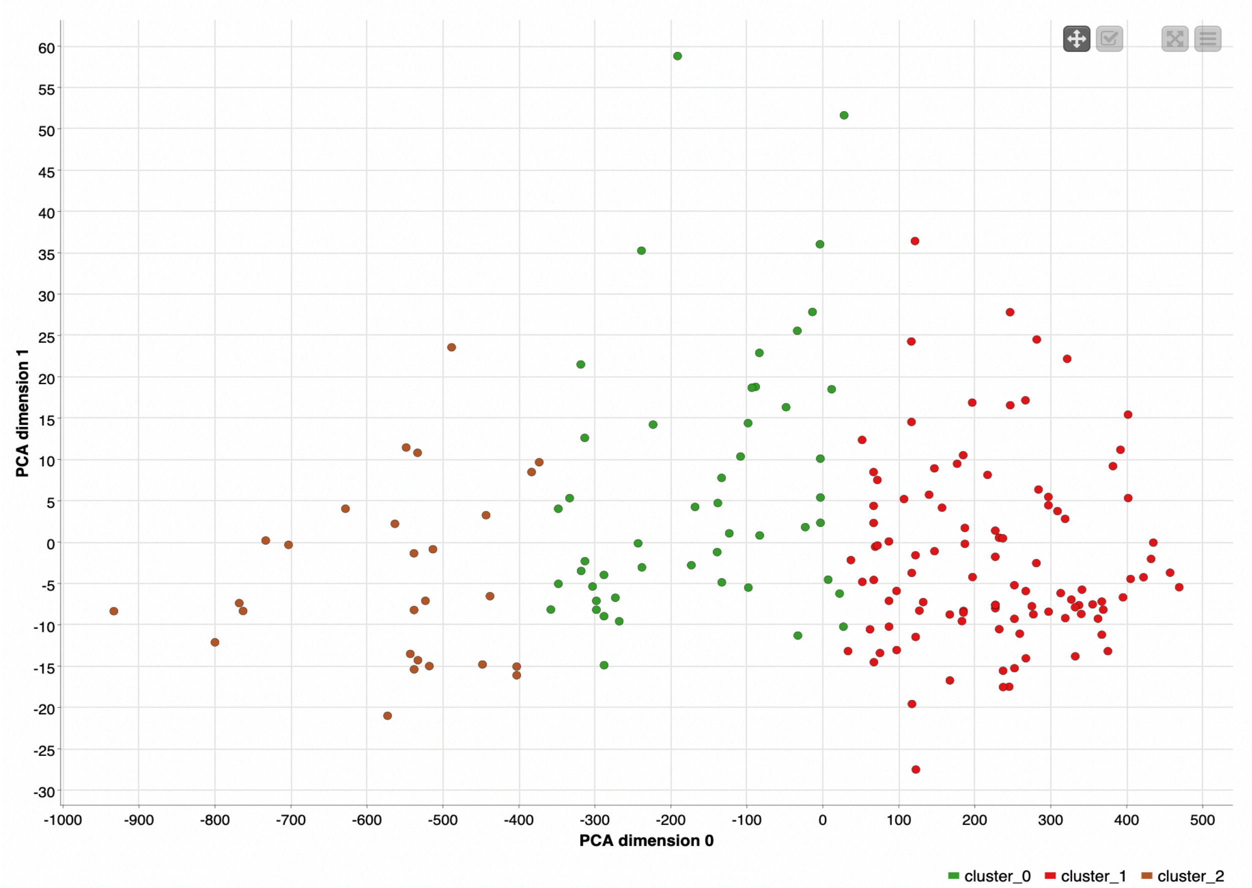


Figure 3: Scatter Plot of Clusters (task 1, plot 2)

Looking at Plot 1 and Plot 2, we can see that class 1 is mainly clustered to cluster 2. However, we observe that many data points are not clustered correctly.

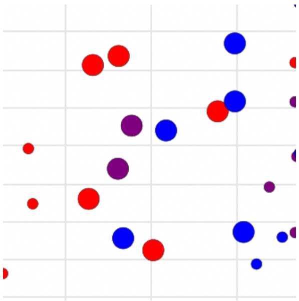


Figure 5

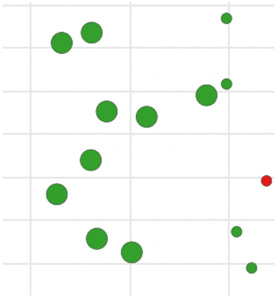


Figure 4

For example, as shown in Figure 4 & 5, the data points belonging to the three classes, were all clustered to cluster 0. The same thing occurs in the second example shown in Figures 6 & 7.

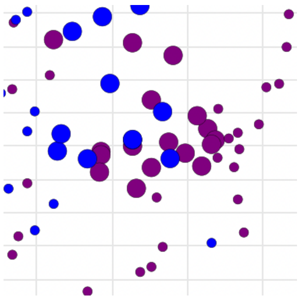


Figure 7

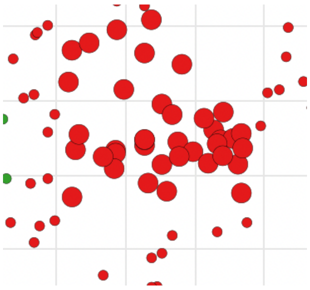
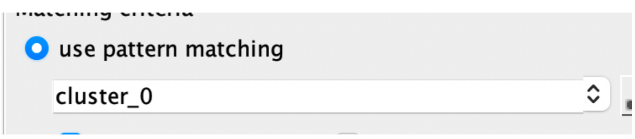


Figure 6

The data points related to classes 2 and 3 (Figure 6), were all clustered to cluster 1 (Figure 7).

To take a closer look at the three clusters, we plot them individually into three scatter plots. This is achieved using the Row Filter node, filtering cluster 0, 1 and 2 each time.



Matching criteria

● use pattern matching

cluster_0

Using the color manager, the three clusters were assigned the same colors used in Plot 2. The scatter plots produced are shown below.

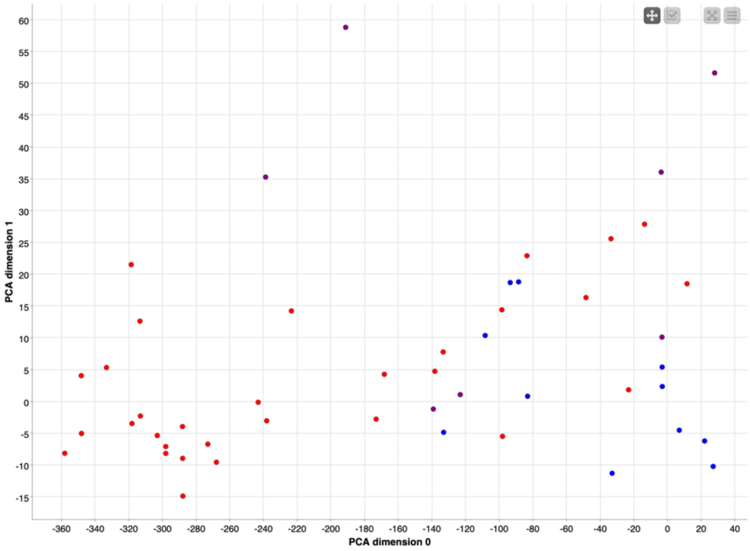


Figure 8: Cluster 0 (Plot 3a)

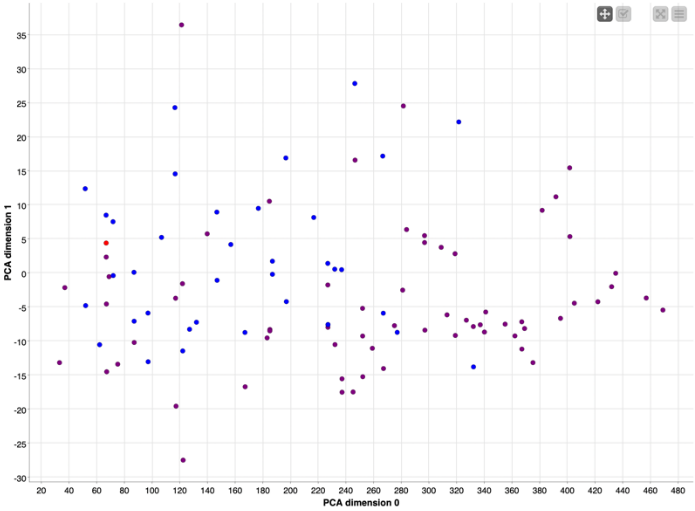


Figure 9: Cluster 1 (Plot 3b)

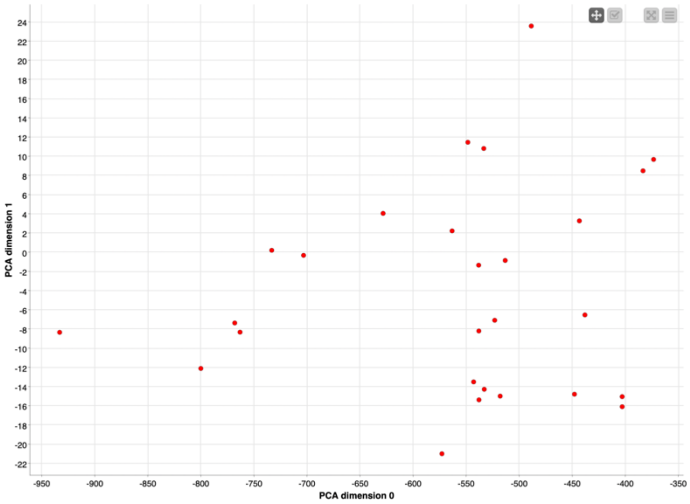


Figure 10: Cluster 2 (Plot 3c)

By observing the above scatter plots, the following can be inferred:

- Cluster 0 consists of a mixture of all three classes
- Cluster 1 consists of a mixture of classes 2 and 3, with a single data point belonging to class 1.
- Cluster 2 consists of data points belonging only to class 1.

To further explore the performance of our clustering algorithms, we look at the following validity measures.

External Index: Used to measure the extent to which cluster labels match externally supplied class labels (Cambridge University Press, 2008).

The following external indices were used:

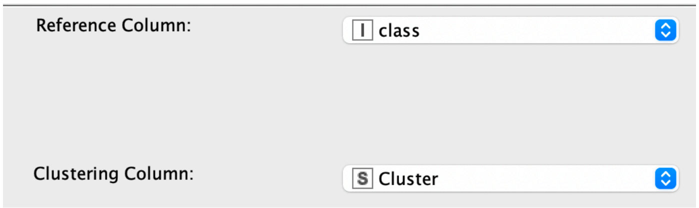
1. Entropy: is a measure that quantifies uncertainty. Entropy decreases as the uncertainty decreases. Entropy (uncertainty) closer to 0, means better clustering.

$$H(p) = -\sum_{i} p_i \log_2 (p_i)$$

where Pi is the probability of the label i (P(i)).

2. Quality: the sum of the weighted qualities of the individual clusters, whereby the quality of a single cluster is calculated as (1 - normalized_entropy). The domain of the quality value is [0,1]. Increased quality means better clustering (KNIME, 2022).

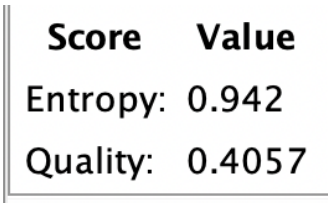
The above statistics were calculated using the Entropy Scorer node which was configured as follows:



Reference Column: class

Clustering Column: Cluster

The results provided by the Entropy Scorer are shown in Figure 11.

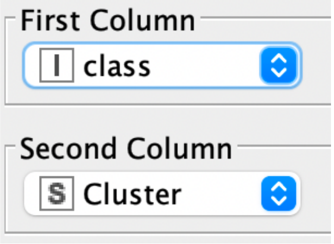


Score	Value
Entropy:	0.942
Quality:	0.4057

Figure 11: Entropy & Quality Validity Measures

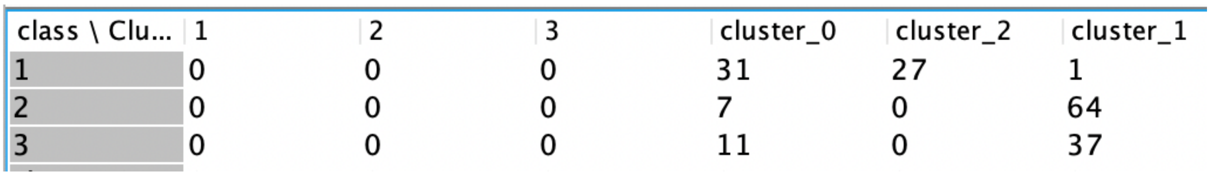
Clustering was performed poorly according to both Entropy and Quality metrics. The quality score is too low, while the entropy score is too high. The high entropy indicates that there is a lot of disorder in the clustering.

To take a better look at how the classes were clustered, we use the Scorer node to see the confusion matrix. The Scorer compares two columns by their attribute value pairs and shows the confusion matrix, i.e., how many rows of which attribute and their classification match (KNIME, 2022). The Scorer is configured to compare the class and cluster columns as shown below.



First Column: class

Second Column: Cluster



class \ Clu...	1	2	3	cluster_0	cluster_2	cluster_1
1	0	0	0	31	27	1
2	0	0	0	7	0	64
3	0	0	0	11	0	37

Figure 12: Task 1 Confusion Matrix

Figure 12 shows the Confusion Matrix produced by the Scorer, which confirms what was observed visually with the Plots 3a, b, c.

Task 2: Clustering with normalization

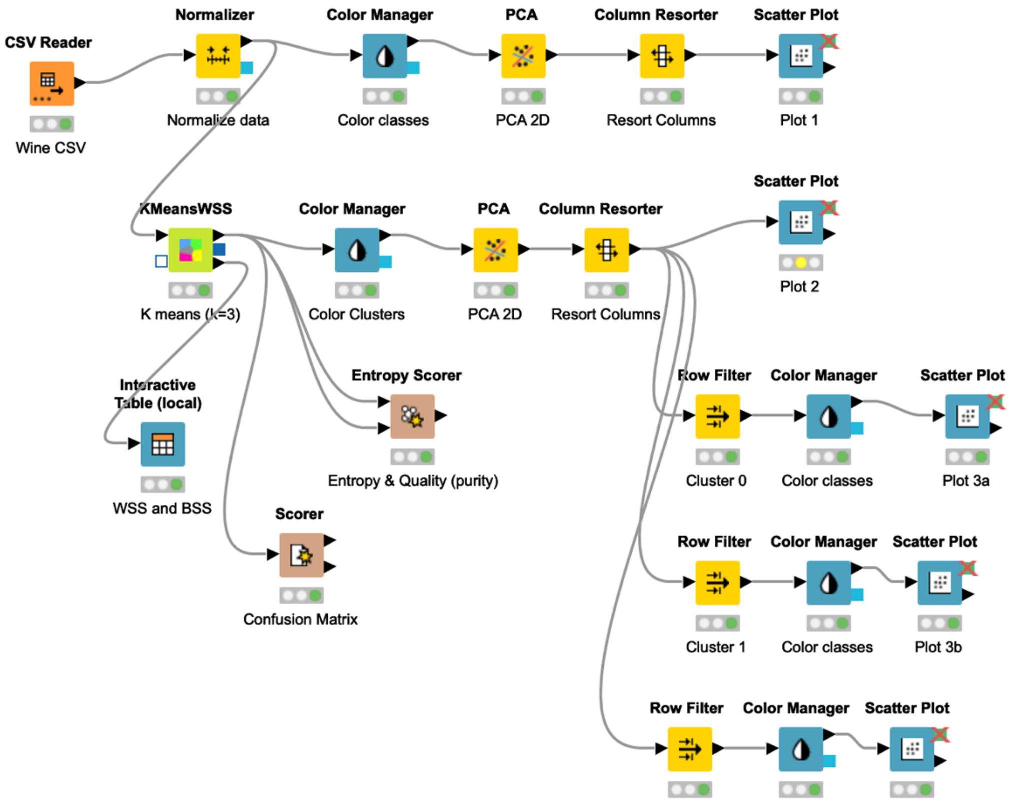
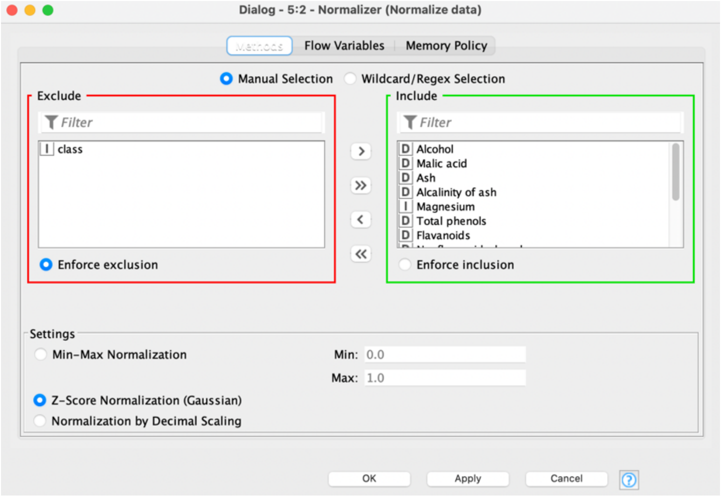


Figure 13: Task 2 Workflow

The workflow for Task 2 is very similar to Task 1, with only difference being that a normalization pre-processing was applied to the dataset. Normalization is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1 (Techopedia.com, 2019). This can be easily implemented in KNIME using the Normalizer node. This node normalizes the values of all (numeric) columns. There are various normalization methods available. The node was configured to use the Z-score normalization (Gaussian) method. Z-score normalization refers to the process of normalizing every value in a dataset such that the mean of all the values is 0 and the standard deviation is 1 (Zach, 2021). The class column is excluded from the normalization.



Dialog - 5:2 - Normalizer (Normalize data)

● Manual Selection ○ Wildcard/Regex Selection

Exclude

I class

● Enforce exclusion

Include

D Alcohol
D Malic acid
D Ash
D Alcalinity of ash
I Magnesium
D Total phenols
D Flavanoids

○ Enforce inclusion

Settings

○ Min-Max Normalization Min: 0.0 Max: 1.0

● Z-Score Normalization (Gaussian)

○ Normalization by Decimal Scaling

The new plots produced are shown in the figures below.

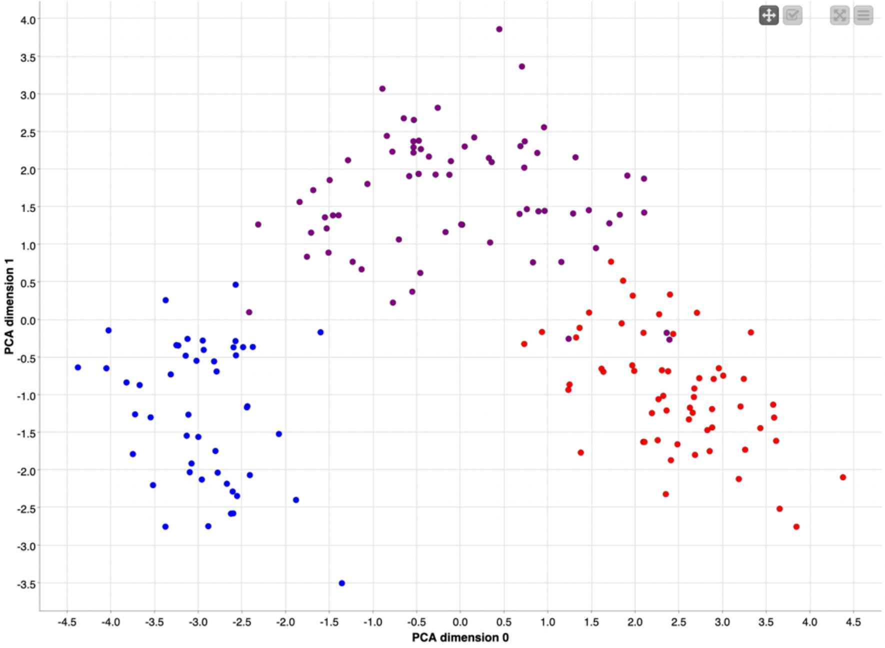


Figure 14: Scatter Plot of Classes (task 2, plot 1)

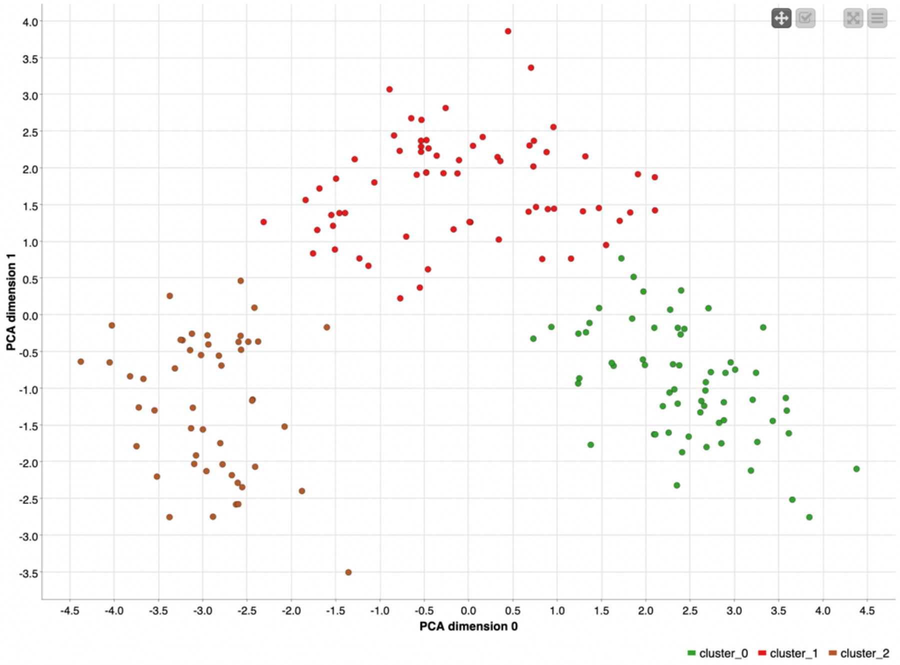
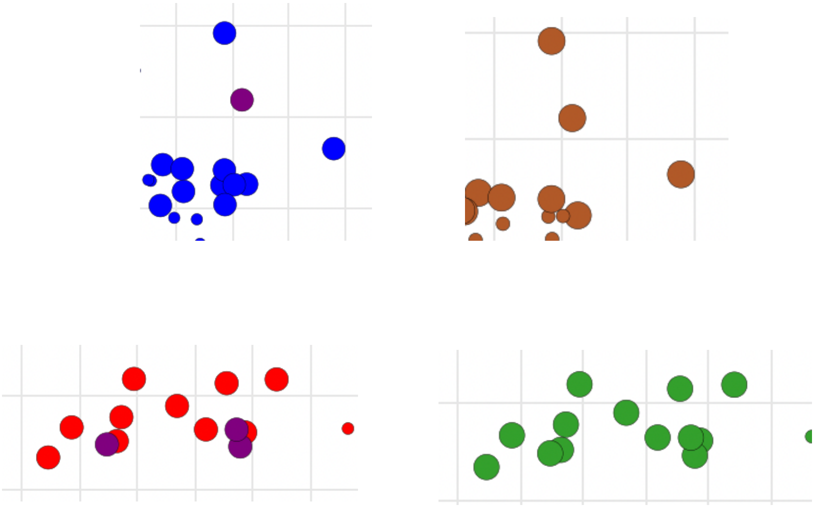


Figure 15: Scatter Plot of Clusters (task 2, plot 2)

The normalized data yields a significantly different plot, as shown in the new Plot 1 (Figure 14), with three distinct groupings of data. The revised Plot 2 (Figure 15) depicts a much clearer distinction between the three suggested classes, which is comparable to the genuine classes. Visibly, there is a small amount of incorrect classification as shown in the data points below.



The clusters were individually plotted again on the following scatter plots.

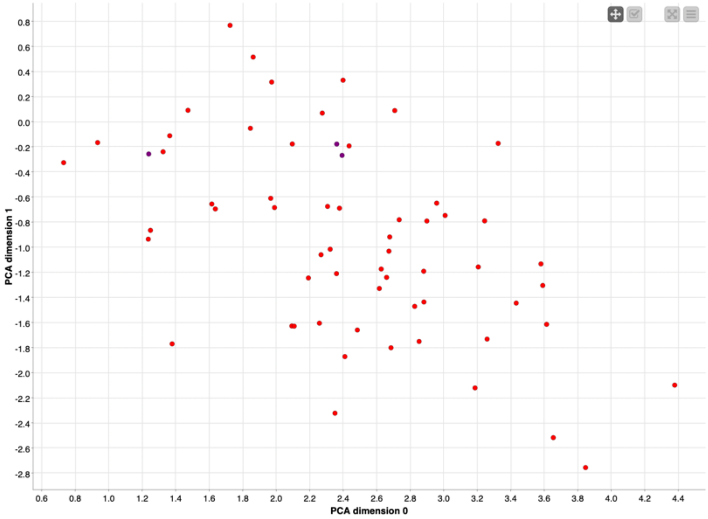


Figure 16: Cluster 0 (Task 2, Plot 3a)

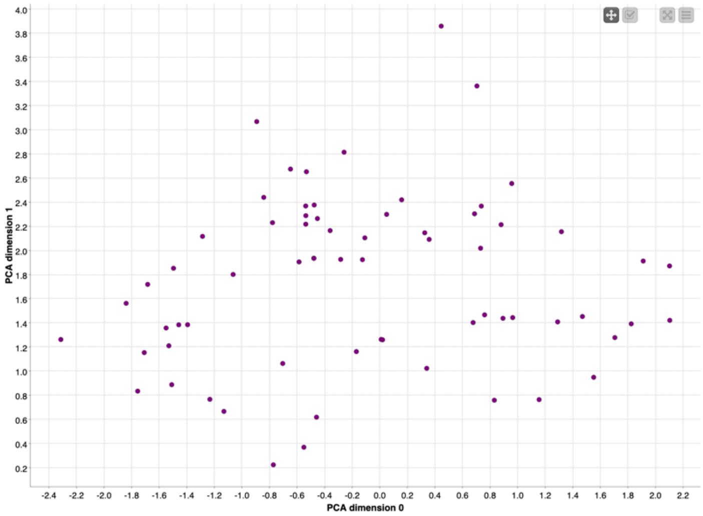


Figure 17: Cluster 1 (Task 2, Plot 3b)

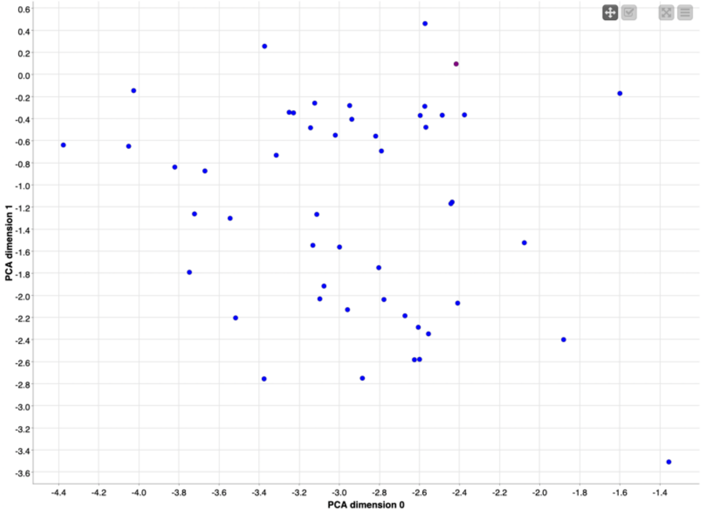
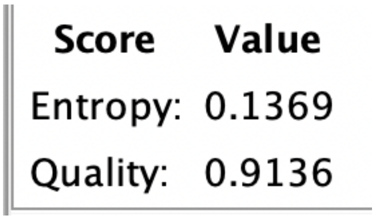


Figure 18: Cluster 2 (Task 2, Plot 3c)

- Cluster 0 displays effective class 1 data point identification, comprising 100% of real class 1 values while also including three data points from class 2.
- Cluster 1 successfully identifies class 2 since it contains 100% genuine class 2 points. However, it does not include all genuine class 2 values as three data points were clustered to Cluster 0.
- Cluster 2 has 100% of the true class 3 values, as well as one class 2 data point.

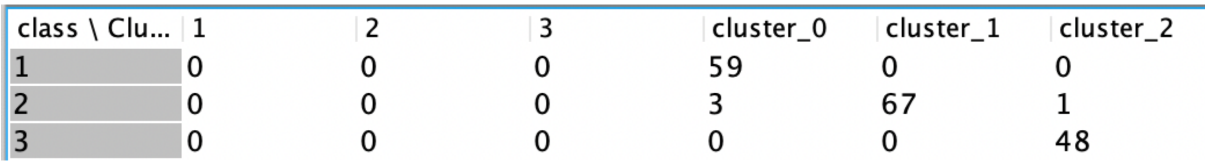
It is apparent that clustering was done considerably more effectively in this situation. We analyze the validity measures once again to learn more about the clustering's performance.



Score	Value
Entropy:	0.1369
Quality:	0.9136

The Entropy Score was significantly improved, as it showed 85.47% decrease, getting very close to zero. The Quality Score was also improved, increasing by 125.19% and its value getting very close to 1 which is the score of an optimal clustering.

The confusion matrix verifies the observation made by looking at the individual scatter plots. Only 4 data points in total were classified incorrectly.



class \ Clu...	1	2	3	cluster_0	cluster_1	cluster_2
1	0	0	0	59	0	0
2	0	0	0	3	67	1
3	0	0	0	0	0	48

Conclusions and Future Work

The results demonstrate that normalization can significantly improve the efficiency of the k-means clustering algorithm. Normalization ensures that redundant data is eliminated and controls the variability of the dataset. The total accuracy of the model might also be calculated to expand this implementation. This would require encoding the Cluster column's categorical values to numbers so that they could be compared to the numerical class column. In their current state, the two columns cannot be compared. As a result, the statistics' accuracy is equivalent to zero. The University of Reading's K-means Clustering node also makes it simple to examine the values of the internal WSS (Within Cluster Sums of Squares) and BSS (Between Cluster Sums of Squares) indices. To make it easy to look at these numbers if needed, the Interactive Table node was added to both Task 1 and Task 2 workflows.

References

www.javatpoint.com. (n.d.). *K-Means Clustering Algorithm - Javatpoint*. [online] Available at: https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning.

Cluster analysis. [online] Cambridge University Press. Available at: https://www.cambridge.org/core/books/abs/cluster-and-classification-techniques-for-the-biosciences/cluster-analysis/FD816B707FACDE9CE9F3BDEFE29703F8 [Accessed 28 Mar. 2022].

KNIME. (n.d.). Documentation. [online] Available at: https://www.knime.com/documentation-3 [Accessed 28 Mar. 2022].

Techopedia.com. (2019). What is Normalization? - Definition from Techopedia. [online] Available at: https://www.techopedia.com/definition/1221/normalization.