# MLCourse-LU

Bart Westhoff 3697932

Assignment 3

## 1 Visualizing clustering results

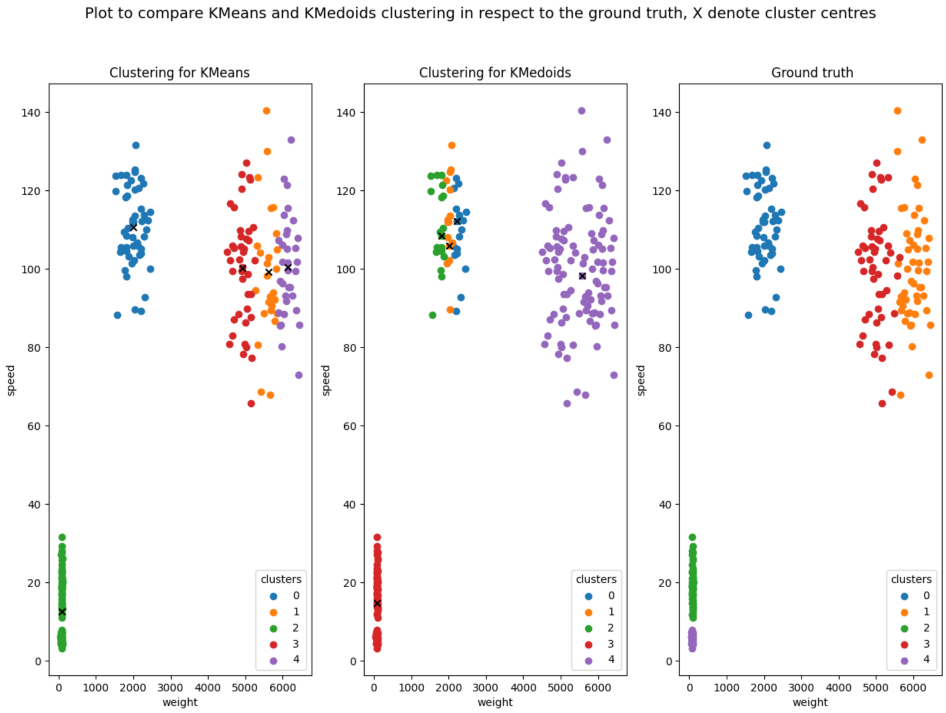

Figure 1: 3 plots showing datapoints of the dataset *vehicles.csv*. It visualizes the differences and common classifications of the 2 cluster methods and their ground truth.

First of all it is very easy to see that both models fail to see the points as two different groups in the lower left. KMeans is better at predicting cluster 1 then KMedoids. But both models fail to do a good job at classifying the clusters.

## 2 $k-$means vs. $k-$medoids

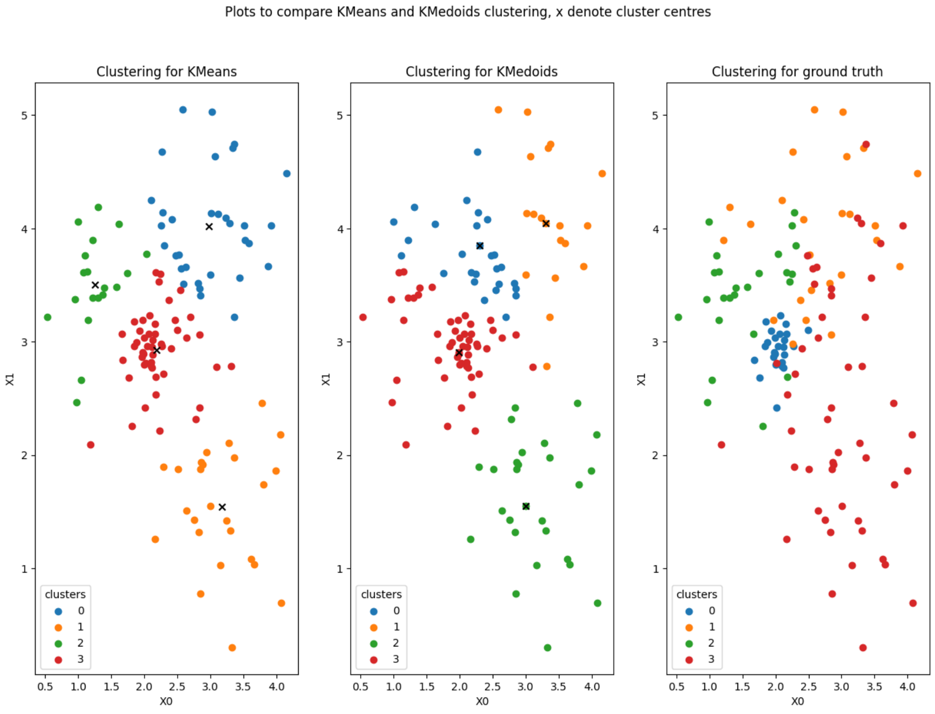

figure 2: 3 plots based on the dataset *dataset-task-2.csv*. The amount of clusters K used is 4 since the ground truth also had 4 labels.

KMeans works better based on the homogeinity and completeness scores. KMeans seems to do better at the upper left corner than KMedoids. However both still fail at recognizing the different clusters in the center.

KMeans peforms much better because the centre of a cluster (noted by the black X) does not have to be on a datapoint. KMedoids will only acknowledge a centre to a given datapoint in the dataset. By this 'imagined' middle point KMeans will calculate the best score on the mean distance between each point.

Kmeans will peform betterin scenario's when there is no real centre in the dataset. Since KMedoids will compare the distance between each datapoint it is more robust between outliers. With Kmeans it will shift the middle point more towards the outlier. Kmeans will peform better when there is no real middlepoint present and when there are less outliers expected. KMedoids will work fine when there are some outliers expected.

| Model | Homogeneity | Completeness | k-value | seed |
|---|---|---|---|---|
| k-means | 0.4169 | 0.4194 | 5 | 42 |
| k-medoids | 0.3728 | 0.3782 | 5 | 42 |

Table 1: Table showing different homogneity and completeness scores based on models and k-values.

## 3 Using the silhouette method to compare $k-$means vs. $k-$medoids on unlabeled data

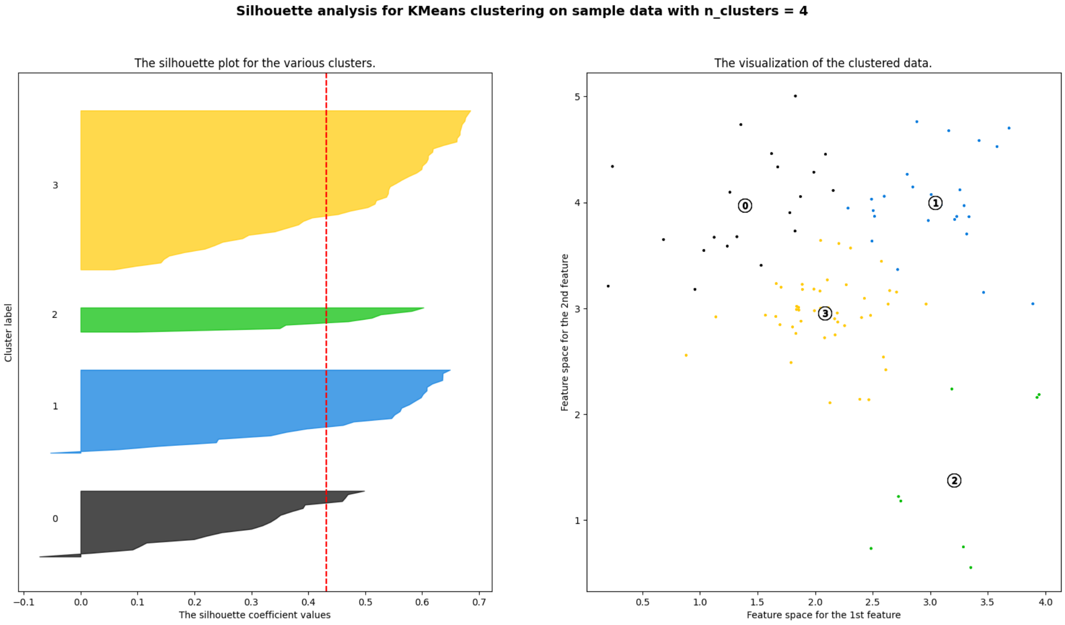

Figure 3: On the left there is a sillhouette plot and on the right a scatter plot of the KMeans algorithm used on *dataset-task-3.csv*.

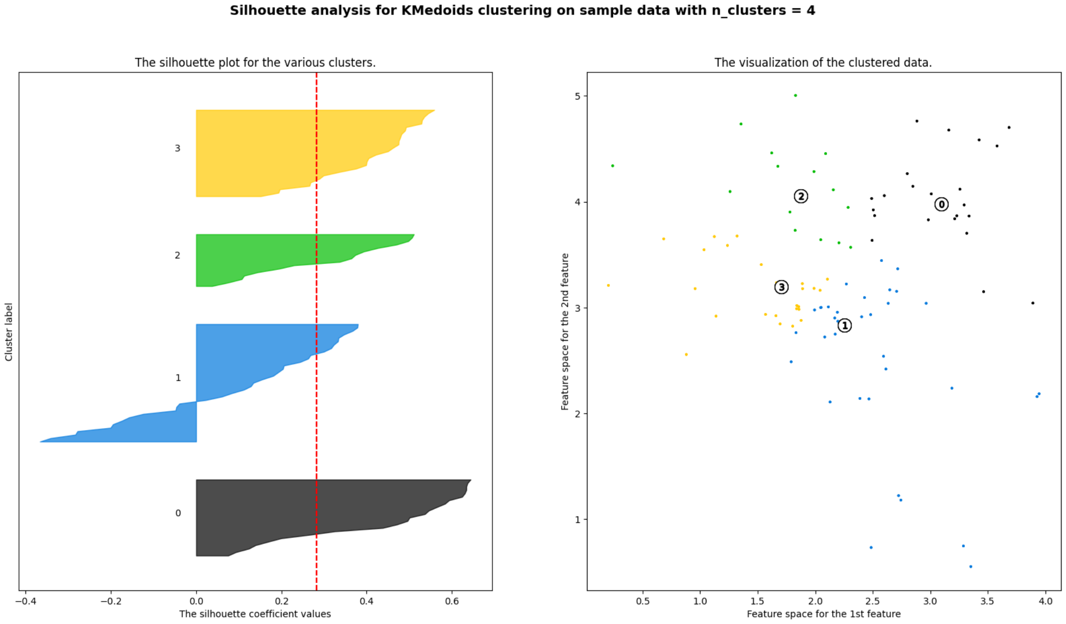

Figure 4: On the left there is a sillhouette plot and on the right a scatter plot of the KMedoids algorithm used on *dataset-task-3.csv*.

| Model | silhouette | k-value | seed |
|---|---|---|---|
| k-means | 0.4323 | 5 | 42 |
| k-medoids | 0.2827 | 5 | 42 |

The silhouette score quantifies the degree of similarity between the points within a cluster and the cluster itself. In this scenario, the KMeans algorithm yields a higher average silhouette score than the KMdoids algorithm, indicating that KMeans outperforms KMedoids. Additionally if you view the scatter plot it intends that the lower right cluster is likely a distinct cluster, which is accurately identified by KMeans due to the larger average distance between any set of points compared to its distance from the central cluster.

# 4 The elbow method

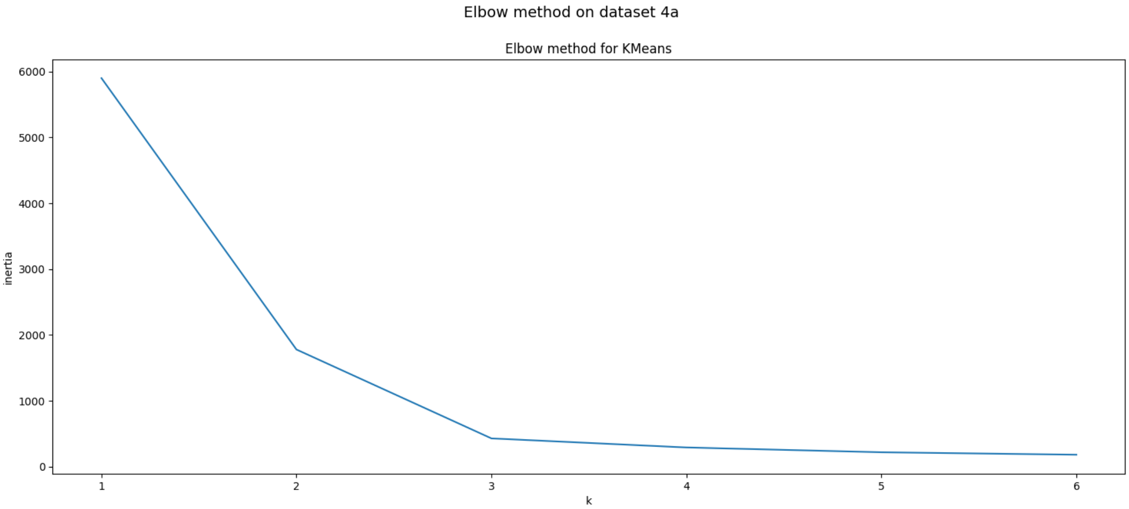

Figure 5: A Elbow plot for KMeans on the dataset generated at exercise 4A

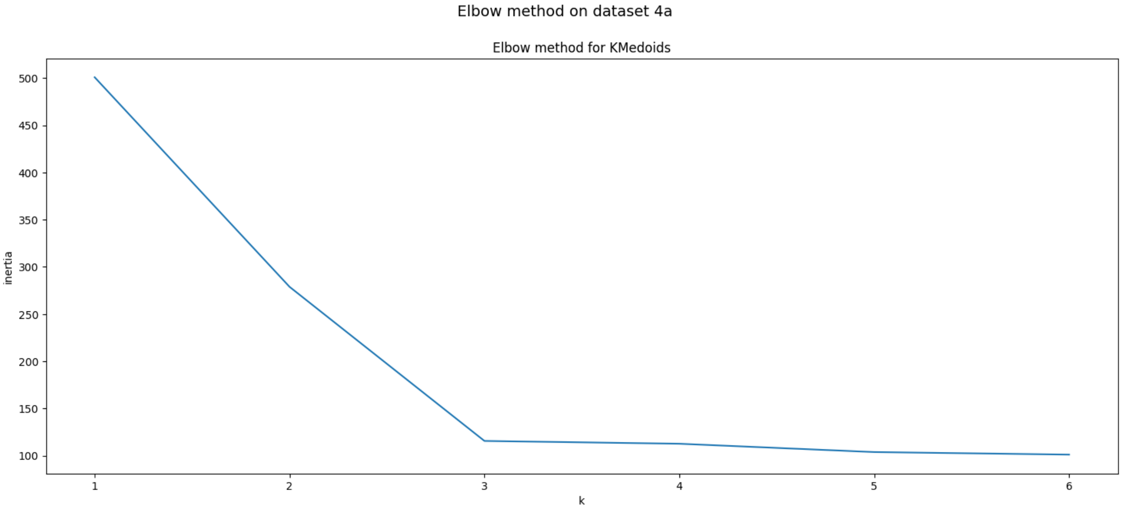

Figure 6: A Elbow plot for KMedoids on the dataset generated at exercise 4A

For dataset 4a there is not much more of a gain after k=3. Both KMeans and KMedoids give this conclusion. Since the decline from k=3 and k=4 is so small it is not beneficial to go for $k > 3$.

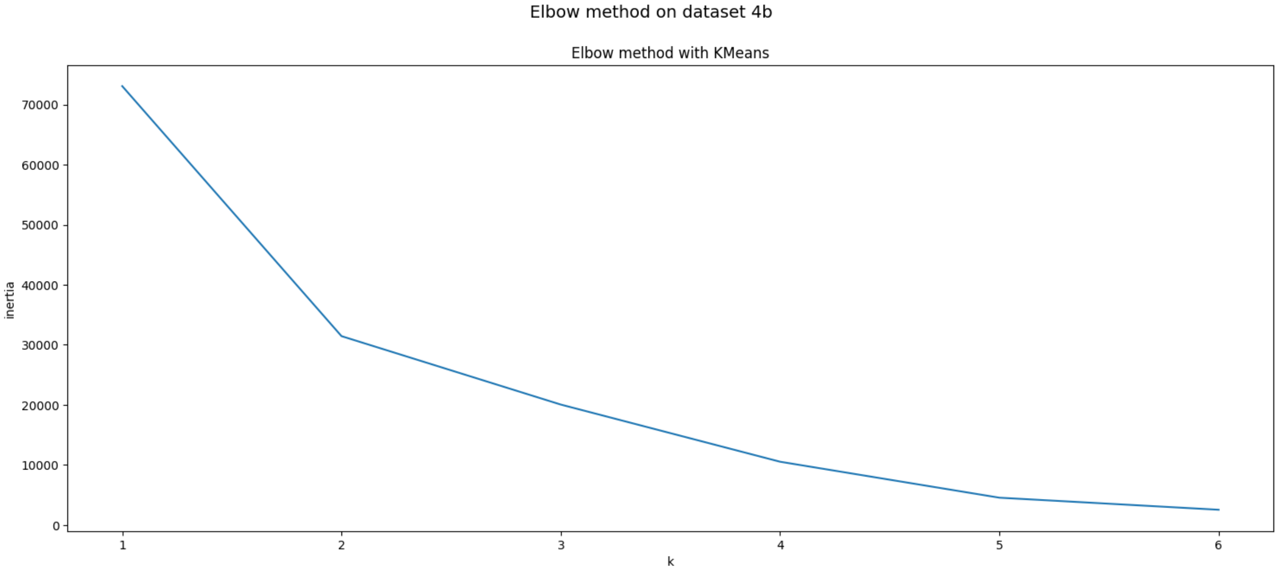

Figure 7: A Elbow plot for KMeans on the dataset generated at exercise 4B

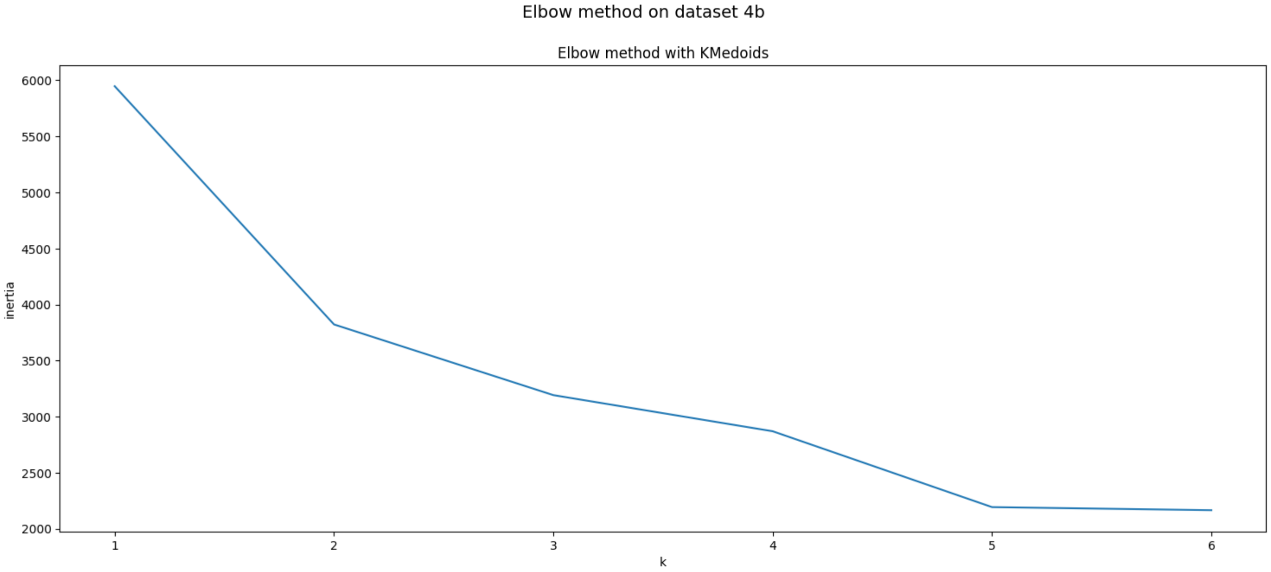

Figure 8: A Elbow plot for KMedoids on the dataset generated at exercise 4B

For dataset 4B I think k=5 will be the best choice. Another choice is k=2 but in my opinion there is too much to win when you choose k=2.

# 5 Generating difficult clusters

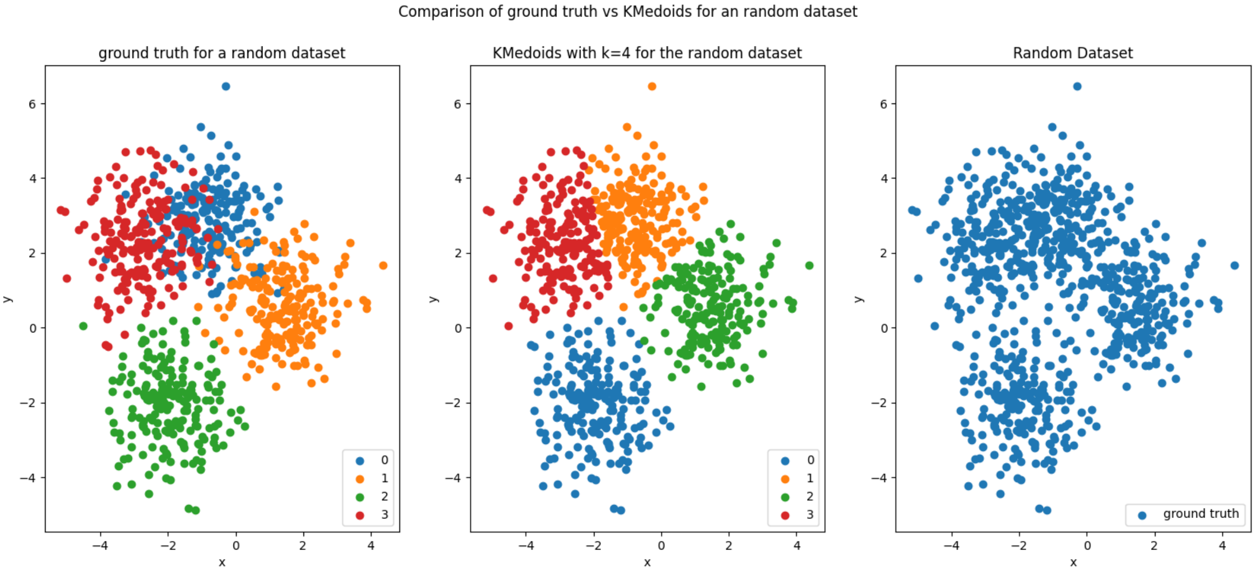

Figure 9: 3 plots to show what a difficult dataset may be for KMedoids to cluster properly. From left to right we have the ground truth, the labels for the clustering algorithm and the random dataset with only 1 color.

The dataset only has two features so it is easy for a human to see the datapoints. The low standard deviation ensures that there will be a packed group of datapoints. When these compact groups overlap it will be harder for the algorithm to distinguish between the ground truth groups. For the human it is still easy to see what the label should be.

Due to the algorithm which selects clusters on similarities it is hard for KMedoids to figure out which points belong to which group since they are all pretty similair. The 4 clusters are as seen close enought to set them apart but for the algorithm it totally does not see the 4th cluster overlap in the first cluster.