QUESTION 1 – REPORT

Martin Labenne

Answer

Methods

The dataset provided had 560 samples, 974 columns, 22 constant features and no null values (all numerical values). Since the means and variances of the features vary substantially, Standardize the data will help a lot for clustering since it keeps the overall shape of the data while putting every feature at the same scale. Even if we are in a n < p situation, it is not necessary to perform variance filtering (we could lose meaningful data, very crud method) because the number of features is still manageable to go further in the process.

Next step is to visualize the data in a low dimension space so that we could estimate the number of groups in the data. To do so, I performed a Principal Component Analysis, as it is simple and convenient way to represent data along high variances axes and clearly see patterns in the data. From *Figure 1* it is obvious that only the three first Principal Components are relevant to explain well data. The *Figure 2* shows a clear segmentation of the data along PC1 and PC2, less on PC3 but still, and then PC4 and PC5 look very gaussian. It seems that including PC4 and PC5 in the model would only add noise, so we decided to drop them. Otherwise, if we focus only on PC1, PC2 and PC3, it is very clear we have 3 or 4 groups of data which look convex (spherical or elongated from the perspective we look at).

Based on those observations I choose 3 different clustering algorithms to try: k-means (top-down approach), gaussian mixture (model based), and hierarchical clustering (bottom-up approach). I thought that it was important to have several approaches to compare the results. Density based Clustering has no utility here, since we do not have any non-convex shapes. Also, since we now are in a 3-dimensional space, we do not have to worry of the “curse of the dimensionality”.

To determine the good number of clusters I used several internal clustering validation measures: the silhouette width index, the Davies-Bouldin Index, and the Clinski-Harabasz Index since they have different definition of what is a good cluster. In addition to those, we looked at one algorithm specific indicator for each clustering method we used (within cluster scatter, Bayesian information Criterion, and the tree height at merging) since I give indications of when the result is optimal for each algorithm. Also, I performed consensus clustering to monitor cluster stability, it ensures that the results are not just because of chance.

Results and discussions

K-means

Since k-means depends on its initial values and perform in a greedy fashion, it is not guaranteed that it will converge to a global optimum. To avoid getting stuck in a local minimum, k-means must run repeatedly. I have done so and only kept the best runs that achieve the minimum within cluster scatter, criterion that k-means tries to minimize. That way, the solution has a better chance to be the global minimum.

I scanned across 2 to 10 clusters. For each cluster count, I performed 1000 iterations: 50 inner k-means iterations (saving the best result) and 20 outer iterations to monitor k-means variability. I used the *k-means++* initialization option, which initializes the centroids to be (generally) distant from each other, leading to probably better results than random initialization. The tolerance was reduced at 0.001 to avoid performance issues, indeed reducing this parameter allows a faster convergence but it reduce precision of the centroid position.

Figure 3 shows the score obtained across 2 to 10 clusters. There are some instabilities above 6 clusters according to the Silhouette width and Davies-Bouldin Index but achieve very high stability overall. Also, these two indexes clearly indicate that 3 clusters are the best fit if we focus on the *low intra-cluster distances and high inter-cluster distances* criterion for clustering (as it achieve respectively maximum and minimum). 2 and 4 clusters have in both cases close scores to the optimum: they are arguably good choices. The within cluster scatter score decreases with the cluster counts and the higher the cluster count is the slower the decrease. We can argue that 3 or 4 are good cluster counts since each time we have a significative decrease (Elbow rule). However, the Calinski-Harabasz Index indicate that 5 (maximum value) is the best cluster count based on *between vs inner cluster dispersion*, but 4 and 6 have similar values. Indeed, the *Figure 6* shows the repartition of 5 clusters across the data, and it is understandable that the score is maximize: with 5 clusters the big group it the middle (PC1, PC2 graph) is cut in half which decrease a lot the overall within cluster dispersion, but also the between cluster dispersion (which explain why 5 is so close to the score for 4 clusters).

On *Figures 4*, *Figure 5*, and *Figure 6* we can observe the results of clustering with k-means and 3, 4, 5 clusters. Visually, 4 cluster clustering makes a lot more sense and is also very coherent with the scores. It provides clean and well separated groups of data.

Gaussian Mixture

As k-means, Gaussian Mixture depends on its initial values to fit the groups of data into gaussian shapes and does not guarantee that it will converge to a global optimum. To avoid getting stuck in a local minimum, we initialized it with k-means. It generally increases cluster selection. Then, the algorithm run and will try to maximize the likelihood of the data given the initial assignments. Repeating this process is guaranteed to always converge to a local optimum. The optimum is achieved if the Bayesian Information Criterion is minimized.

As with k-means, I scanned across 2 to 10 components. For each cluster count, we performed 1000 iterations: 50 inner Gaussian Mixture Model iterations (saving the best result) and 20 outer iterations so we can monitor the variability. We allowed each component to have its own general covariance matrix to achieve a better precision in clustering. It is a 3-dimensional space, so it is not a big deal for performances.

From *Figure 7*, it is understandable that Gaussian Mixture Model leads to very unstable results in every case except for 3 or 4 clusters. 2 is a very unstable choice and most of the times it provides a poor result compared to other choices. The elbow on the Bayesian Information Criterion score is clearly indicating us that 4 is the optimal result for this algorithm and settings. Calinski-Harabasz index is maximized with a 5 cluster counts (at least 50%) of the time but lead to unstable results. Having 4 clusters leads to a perfectly stable and very high score.

Plotting the data results in pair plots extremely like k-means, so I decided to not include them in the report. However, it is not surprising that the result is very similar since the data groups are convex and very well separated. The scores and plots are clear indications that if the data has a gaussian behavior, then 4 is the best cluster count.

Hierarchical Clustering

I repeated the clustering for hierarchical clustering. Given a dataset, the choice of a linkage and the choice of a metric, the clustering is effectively deterministic. This way, it is enough to compute the tree one and investigate it for different cluster counts. I used the Ward since it is the most suitable for our problem: it minimizes the sum of squared difference within all clusters, it works well with convex data and is the most robust linkage to noisy data.

From *Figure 8*, the scores are very similar to the k-means' ones, which is not very surprising because Ward is a variance-minimizing approach and is like the k-means objective function but tackled with an agglomerative hierarchical approach. From Silhouette width, Davies-Bouldin Index and Calinski-Harabasz Index, the consensus goes toward a cluster count of 4, which is supported by the fourth graph of the figure. Indeed, the tree height at merging represents the value of the linkage function when two clusters are merged. Big jumps represent high cost which generally means well separated clusters joint together. That big jump is clearly between 3 and 4 clusters, which indicates that a good cut would be at 4 clusters.

Plotting the data results, once again, in pair plots extremely like k-means, so I decided to not include them in the report and once again, 4 clusters is the choice that provide the cleanest result.

Consensus Clustering

Since the 3 algorithms had the same clustering results, I decided to only perform consensus clustering with k-means only. I randomly subsampled the data 50 times and perform k-means (with 10 inner iterations, which still gives very precise results) for each cluster count within 2 and 10. I repeated the experiment 5 times to measure the uncertainty and reported the result in a graphical way.

The first thing I noticed from *Figure 9* is that there almost no uncertainty in the consensus here which is not surprising since k-mean was very consistent so far in this exercise. Also, for 2, 3 or 4 clusters the algorithm is very stable since the lines are exactly flat from the 0.01 quantile and the 0.99 quantile. The results for 5 clusters and more are less stable since it shows some partitions between 0 and 1.

Conclusion

Since the result of 4 clusters is the best looking from 3 different models that fit the problem, we can say with the certitude of the Consensus Clustering that 4 is the best cluster count to fit the data in cluster dependent algorithms (like k-means). The visualization of the data should look like *Figure 5*.

Figures

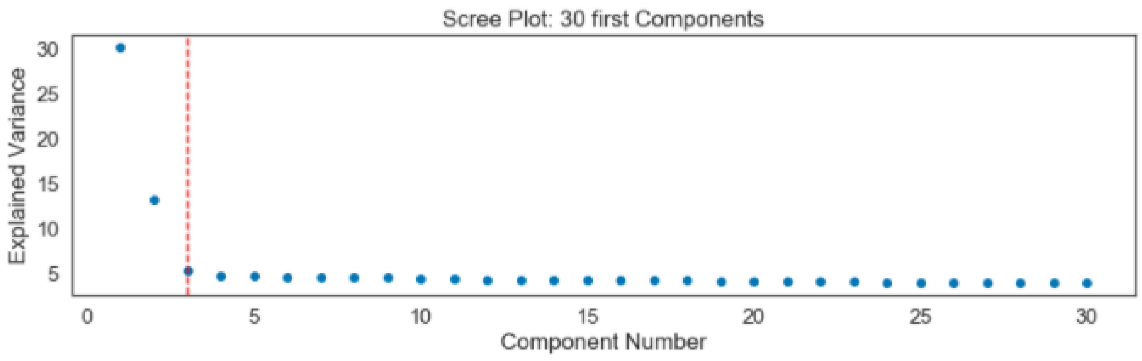

Figure 1: Scree plot of PCA performed on Standardized dataset.

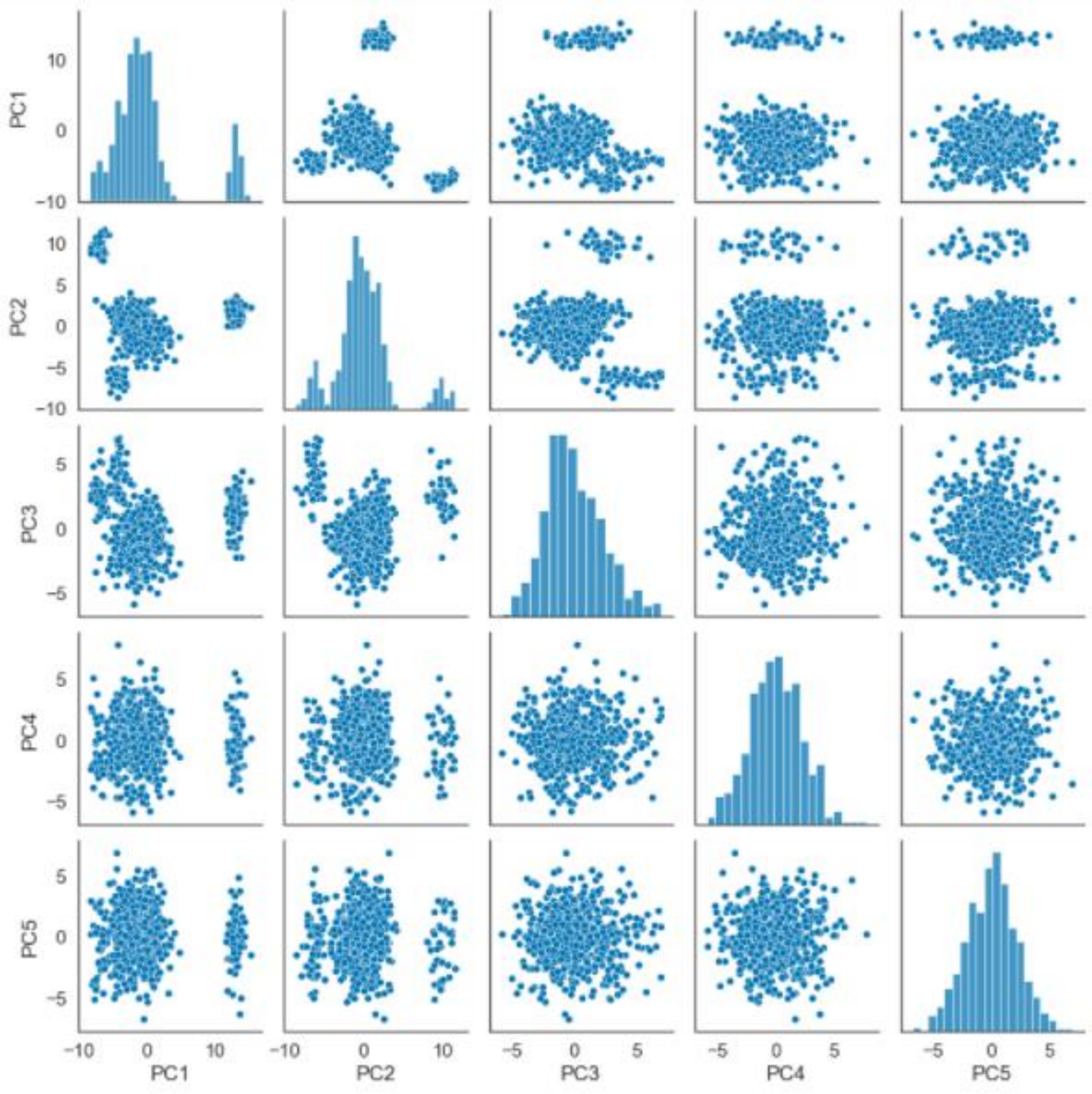

Figure 2: PairPlot of the 5 first Principal Components

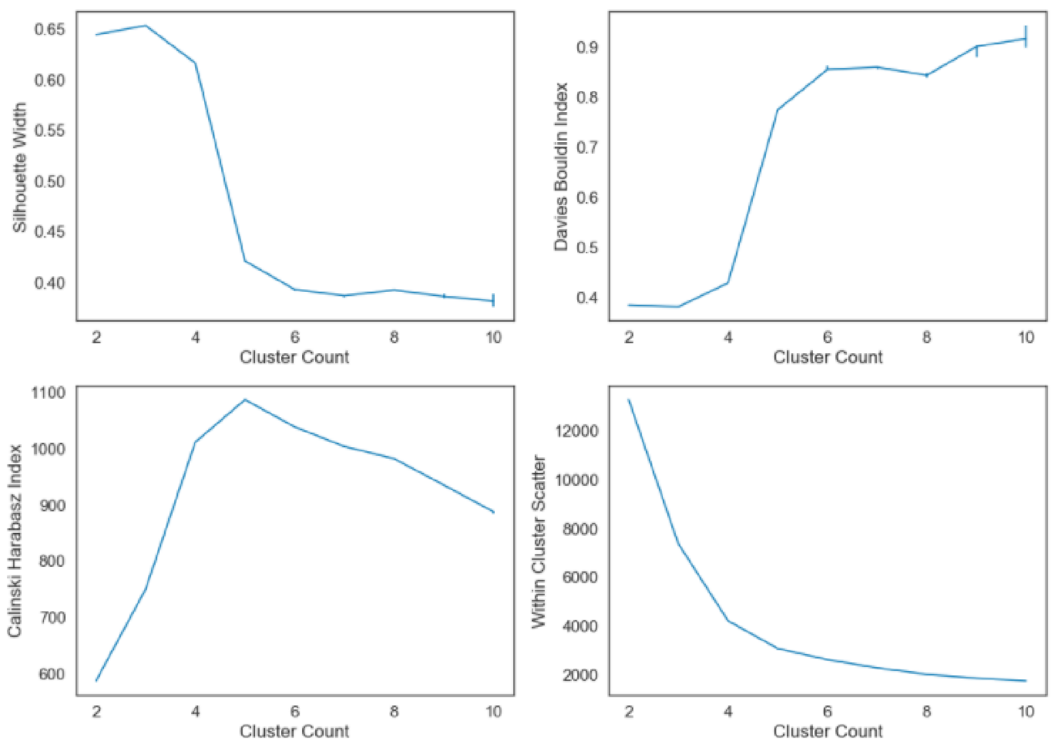

Figure 3: Scores for k-means clustering, the solid lines show the median value of the indices across the number of outer runs. The bars show the range of values achieved for that cluster count.

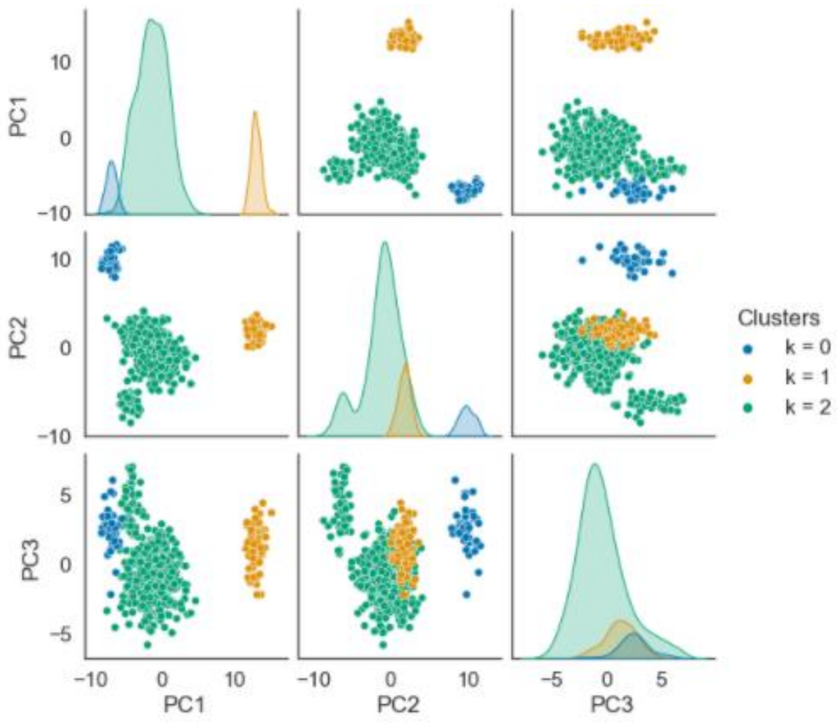

Figure 4: Result of clustering with k-means with 3 clusters

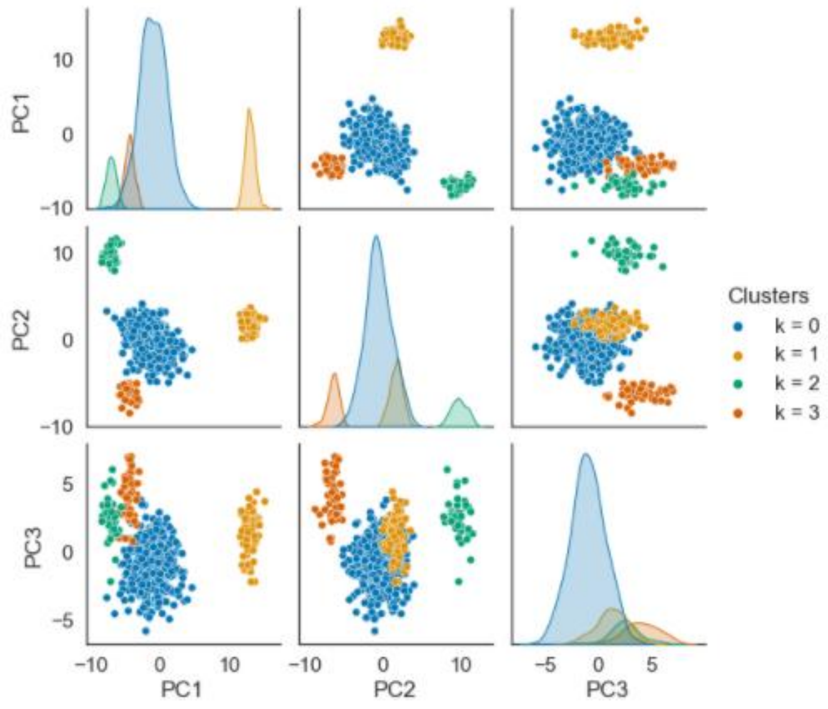

Figure 5: Result of clustering with k-means with 4 clusters

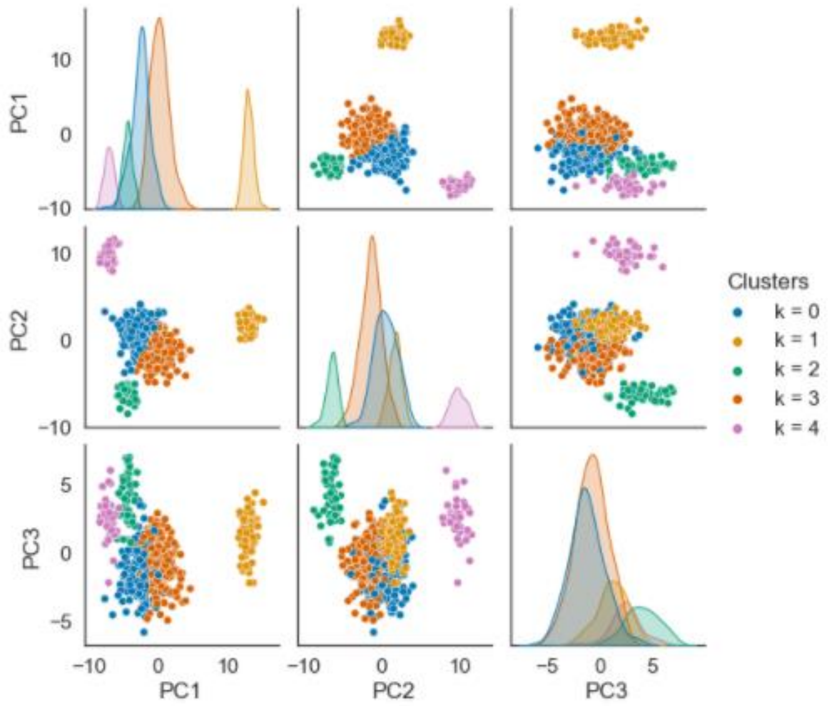

Figure 6: Result of clustering with k-means with 5 clusters

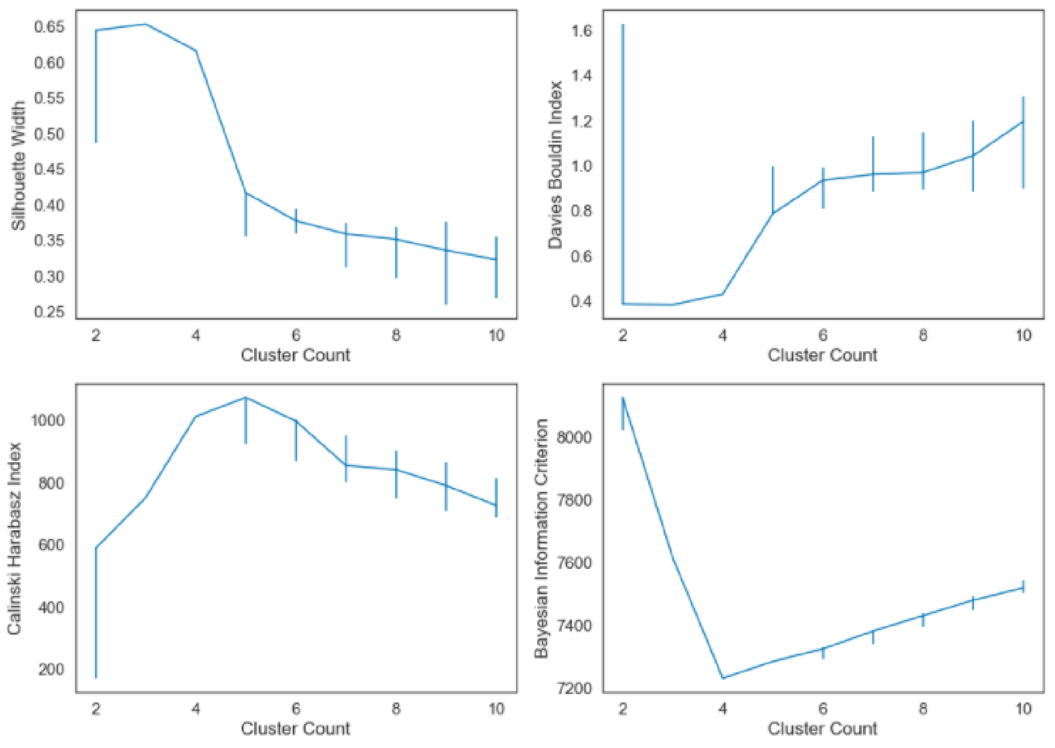

Figure 7: Scores for Gaussian Mixture clustering. The solid lines show the median value of the indices across the number of outer runs. The bars show the range of values achieved for that cluster count.

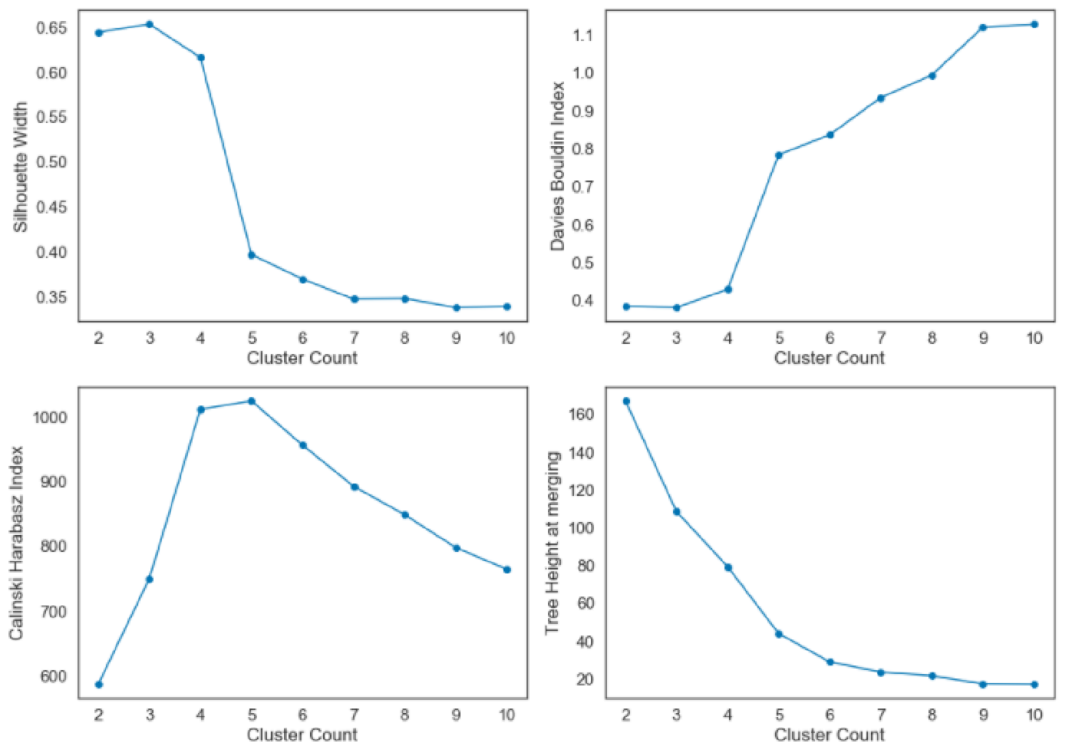

Figure 8: Scores for Hierarchical clustering with Ward Linkage. The ponits represent the value of the indices across the number of clusters. There no measure of incertainty here since the algorithme is deterministic.

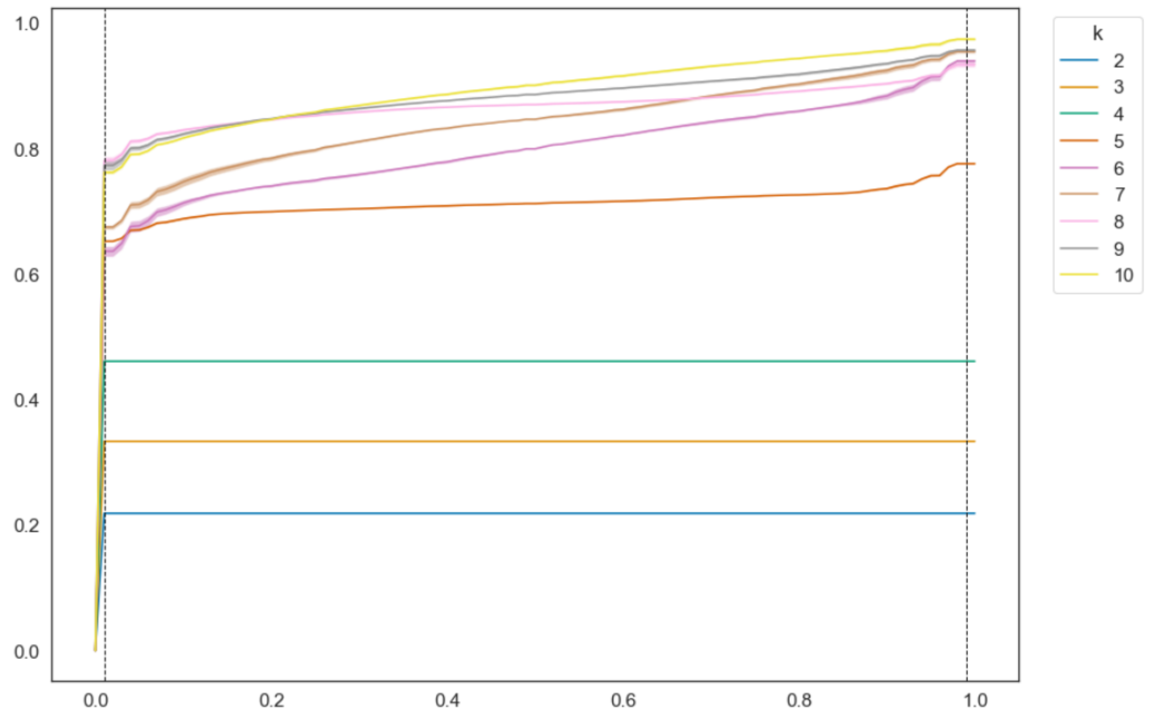

Figure 9: Empirical distribution of flatten concessus matrices for each cluster count. Dash lines represente the 0.01 quantile and the 0.99 quantile.