Quantization-Based Clustering Algorithm

Bernat Martínez

May 2020

1 Abstract

The goal of this practical work is to implement the quanitization based clustering algorithm based on the original paper. After implementing some experimentation using artificial and real datasets is done in order to compare it to some well known clustering algorithms. In this case I will focus on the K-Means [4] family of algorithms since they work on a similar manner, the main type of optimization on this family of algorithms usually focuses on the distance computations necessary. This algorithm tackles such problem by quantizing data in a multidimensional histogram, this enables a series of properties that greatly reduce the computations necessary.

2 Introduction

Clustering is a widely used technique which enables a lot of useful tasks, for example knowledge discovery, data mining and image segmentation. A lot of research efforts have been used in trying to find improvements and optimizations to this problem, with a very large sample of data the majority of algorithms lose effectiveness.

There are multiple approaches to this problem that can be separated on different families, for example density based or hierarchy based methods. However, in this case the main focus is the K-Means family, which is based on trying to optimize a Voronoi partition (figure 1) in a greedy manner. The algorithm on the chosen paper, which is called Quantization Based Clustering algorithm, proposes an approach that can considerably reduce the number of distance computations necessary in order to process large sets of data without losing quality.

The proposed approach is subdivided in three different phases, firstly all the data is quantized and assigned to histogram bins. Secondly the quantized data is exploited to perform clustering at an histogram bin level, after that each data point is assigned to the corresponding cluster. The differentiating idea in this case is that clustering is performed in the quantized version which is represented

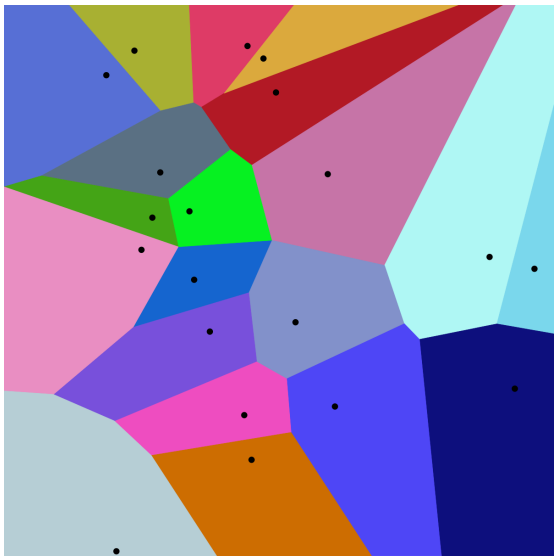

Figure 1: Voronoi partition example

by the multidimensional histogram.

3 QBCA

In this section the main steps of the QBCA [3] algorithm implementation will be explained, mathematical demonstrations of the different optimizations can be seen in great detail on the original paper.

The clustering process consists on a series of steps that can bee seen at figure 2. Firstly all of the data points are translated to the quantized paradigm, after that the cluster center initialization occurs where the algorithm tries to find some good initial cluster centers. After the centers have been initially positioned the main step occurs which is the cluster center assignment using the quantized data, here is where the main optimization of this methodology occurs. After that the cluster centers are recomputed and the cluster center assignment is repeated, these last two steps are repeated until the ending criteria is met.

In this case the ending criteria consists on the sum of the distances between each cluster center and their previous position, if the difference is lower than a defined value the algorithm terminates.

3.1 Quantization

To perform quantization two parameters need to be computed, the first one which is called ρ determines how many histogram bins we will use on each

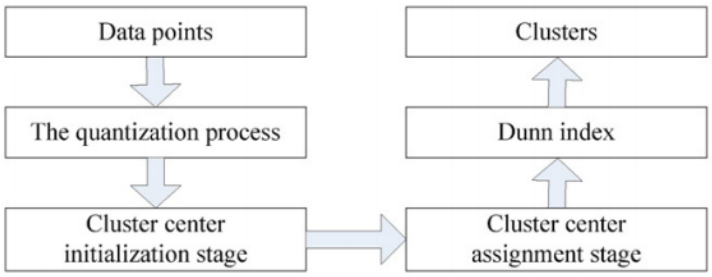

Figure 2: Qbca pipeline steps diagram.

dimension. The second one called λ is a vector determining the size of the histogram bins in each different dimension. To compute them the following equations are used:

$$\rho = \lfloor log_m n \rfloor \tag{1}$$

$$\lambda_l = \frac{\bar{p}_l - \underline{p}_l}{\rho} \forall l \in \{1, ..., m\} \tag{2}$$

Where m is the number of dimensions and n is the number of data points, $\bar{p}_l$ and $\underline{p}_l$ refer to the maximum and minimum value on the l-th dimension. After the basic parameters needed to perform the quantization are computed we must know how does the quantization function work. The output of this function will be an index referring the position of its corresponding histogram bin. The quantization function works with the following formulas:

$$\mathbf{b}.id = \sum_{l \in \{1,...,m\}} (\xi_l - 1)\rho^{m-l} + \xi_m \tag{3}$$

$$\xi_l = \begin{cases} \lceil p_l - \underline{p}_l \rceil & if p_l > \underline{p}_l \\ 1 & if p_l = \underline{p}_l \end{cases} \tag{4}$$

The obtained $\mathbf{b}.id$ is the linearized index of the m-dimensional histogram.

3.2 Cluster Center Initialization

At this stage we already have the quantized representation of the data, the next step is to compute some good initial cluster centers, also named as seeds on the original paper. The motivation behind the proposed initialization is based on how Gaussian distributions work, in such distribution the closest region of the center is the one containing higher number of points.

In order to perform the seed search efficiently a max heap is utilized. A max heap is a type of tree structure in which the element always stands on the root node. Lets assume that all the data has already been quanized and we have the cont of data samples on each bins. The algorithm process is the following:

1. Initialize a Max Heap H with the non zero bins using their count as criteria and a seed list L;
2. While H is not empty pop the top of H and check if it has the maximum count of all its neighbouring histograms, in such case add it to L
3. If L contains a number smaller than the desired seed number include enough of the remaining bins with the largest counts that where not already on L.
4. If L is bigger than the desired number of seeds select the k candidates with the largest counts
5. With the final k seeds already selected compute the real center value, in order to do so compute the center using the samples on the chosen bin as a cluster, this center is the seed.

3.3 Cluster Center Assignment

In the previous step the initial seeds where computed, now its time to assign each data point to its respective cluster. To do so in an efficient manner the algorithm utilizes a property that allows it to greatly reduce the amount of distance computations. In order to use the algorithm the maximum and minimum distances between seeds and histogram bins must be computable in the space $\mathbb{R}^m$. To understand how the distance computations for a given seed s_h and a concrete bin b_j are made look at the following formulas, the first one is used to compute the minimum distance:

$$\underline{d}(s_h, b_j) = \sqrt{\sum_{l=1}^{m} (s_h - b_{j,l})^2} \tag{5}$$

$$b_{j,l} = \begin{cases} \underline{b}_{j,l} & if s_{j,l} < \underline{b}_{j,l} \\ \bar{b}_{j,l} & if s_{j,l} > \bar{b}_{j,l} \\ s_{h,l} & otherwise \end{cases} \tag{6}$$

where $b_{j,l}$ is the value of the histogram bin at dimension l, $\underline{b}_{j,l}$ and $\bar{b}_{j,l}$ are the minimum and maximum coordinates of the given histogram in the l-th dimension.

To compute the minimum distance between a seed and an histogram bin a different formula is used:

$$\bar{d}(s_h, b_j) = \sqrt{\sum_{l=1}^{m} (s_h - \lambda_{j,l})^2} \tag{7}$$

$$\lambda_{j,l} = \begin{cases} \underline{b}_{j,l} & if s_{h,l} \geq \dfrac{\underline{b}_{j,l} + \bar{b}_{j,l}}{2} \\ \bar{b}_{j,l} & otherwise \end{cases} \tag{8}$$

This definition of the distances allows us to make use of the **lemma 1** explained on the original paper in which if $\underline{d}(s', b) \leq \bar{d}^*(s, b)$ is satisfied then we can confirm that s' is not the closest of any point p belonging to the bin b. Computing $\bar{d}^*(s, b)$ is quite easy, just do the following:

$$\bar{d}^*(s, b) = \min_{1 \leq h \leq k} \bar{d}(s_h, b) \tag{9}$$

All seeds that match this condition for a specific histogram bin it can be considered a candidate for it. Knowing how the distance computations work internally we can proceed to show how the cluster center assignment works in a few simplified steps, for each non empty histogram bin bj do the following:

1. Calculate the maximum distance $\bar{d}$ between the bin and all of the seeds.
2. Sort the maximum distances obtained in ascending order and take the first one, this value is $\bar{d}^*(s, b)$
3. For each seed check if the condition stated in lemma 1 is met and keep the ones that satisfy it as candidates of the given bin.
4. For each point inside the bin find the minimum distance from all of the candidate seeds and assign p to its closest, this is done in the $\mathbb{R}^m$ paradigm.

As it can bee seen from the explanation the final distance computation only has a few seeds to take into account (candidates) because the others have already been discarded.

3.3.1 Centroid Recomputation

After the cluster assignment phase ends the cluster centers must be updated, this is done in exactly the same way as standard k-means by using the points of each cluster and their values in $\mathbb{R}^m$.

4 Experimentation

This section contains information regarding the data used and which experiments have been done with it. The main focus of the experimentation is to check the performance difference and clustering quality with both an internal and an external index.

4.1 Data

There are four datasets including three different origins being used for the experimentation process. These are the datasets being used:

- **Synthetic Data:** this dataset [1] contains four distinguishable clusters that come with their respective labels, it is only 2 dimensional and data attributes only contain the points positions. It has a sample size of 1000.

- **Satimage:** this is an already well known dataset [2] it consist of 36 attributes plus their class labels. The size of the dataset is 6430 instances, representing multi-spectral values in 3x3 pixel groups from a satellite image. The database is a (tiny) sub-area of a scene, consisting of 82 x 100 pixels. Each line of data corresponds to a 3x3 square neighbourhood of pixels completely contained within the 82x100 sub-area. Each line contains the pixel values in the four spectral bands (converted to ASCII) of each of the 9 pixels in the 3x3 neighbourhood and a number indicating the classification label of the central pixel.

- **Images:** Two images have been randomly selected from online sources in order to perform clustering on them for image segmentation. The images loaded are resized to a smaller height of 150 pixels while keeping their aspect ratio. RGB and pixel coordinates are used for the segmentation without any type of weight difference between them.

4.2 Performance analysis

The two main things that have to be tested when experimenting are both the number of distance computations and the quality of the clusters. Computation time would also be a good performance indicator, but despite the algorithm implementation being quite fast it cant compete with the Sklearn MMeans which is heavily optimized. However, distance computations are a very good way to infer how many computation time can be gained with the algorithm.

In order to check the quality of the clusters we use one an intrinsic index, this type of index doesn't use any type of ground-truth and focuses on the quality of the clusters themselves. The index used in this case is the silhouette score [5] included in the Sklearn library The results are extracted by computing the mean results obtained from 10 repetitions of each configuration, except in the case of the visual comparisons. The configurations used are a fixed number of dimensions that varies depending on the dataset and a range of two to five different cluster centers. For each one of the experiments we provide a quantitative and a qualitative comparisons The qualitative comparison is given with the best performing configuration so it is easily observable if there are clear differences on the results obtained.

4.3 Artificial Data

Firstly we extracted results for the artificial dataset in which four different clusters are easily identifiable. By looking at figure 4 the silhouette indexes obtained can be seen using from two to five centroids (seeds in QBCA). For this dataset we can see that there is no noticeable quality loss, for five and two clusters is a bit lower but for four is higher. This small difference is probably due to variance.

The fact that there is not a consistent performance decrease in the quality of the clusters generated is already a very good indicator, but for the algorithm

Figure 3: Distance computations on test dataset with 2 dimensions.

to accomplish its purpose this has to come with a smaller number of distance computations. If we observe figure 3 this can be easily confirmed. For example, with the best performing number of centers (which is four), we can see that both algorithms converge pretty easily in comparison to the other runs. In this specific case K-Means performs 4000 distance computations in a dataset of 1000 samples, however QBCA heavily outperforms it in that aspect and performs only 2000 computations.

The improvement is not always the same, depending on the situation the quantizatiion properties are exploited less efficiently and the reduction becomes smaller. At the results obtained with 5 clusters we can see that both algorithms had a harder time converging and the performance improvement is not so noticeable, nonetheless it is still an improvement in terms of this metric.

To have a qualitative evaluation one of the ten runs performed using four clusters has been plotted at figure 5, given the shape of the data in the two dimensional space it becomes obvious that this was the number that led to better results. In terms of the final clusters no visually distinguishable difference can be deduced, the cluster colours are different due to different order but the final meaning is the same.

4.4 Real Dataset

It has already been confirmed that the algorithm leads to a considerable improvement in terms of distance computations, however, to have more significant results we have to test the algorithms in a wider range of cases. In this second batch of experiment the Satimage dataset has been used as mentioned in previous sections. This is a well known dataset that serves as a good benchmark for this task.

The original dataset consists of 36 different dimensions, to make the data more tractable it is reduced to an arbitrary number of five dimensions using

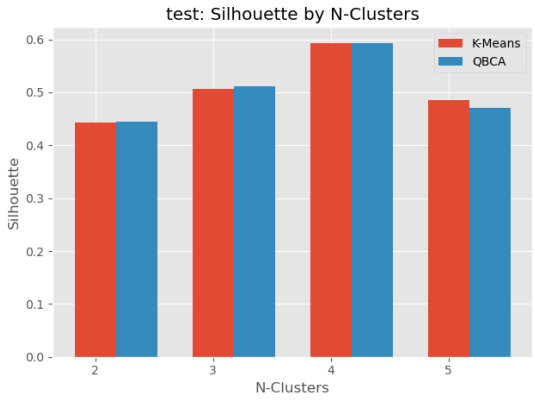

Figure 4: Silhouette scores on test dataset with 2 dimensions.

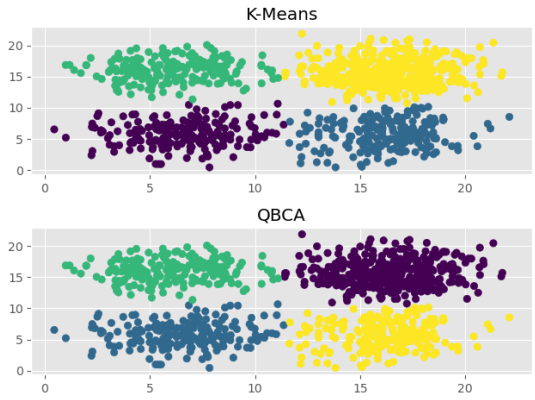

Figure 5: Clustering results on test dataset with 2 dimensions and 4 seeds.

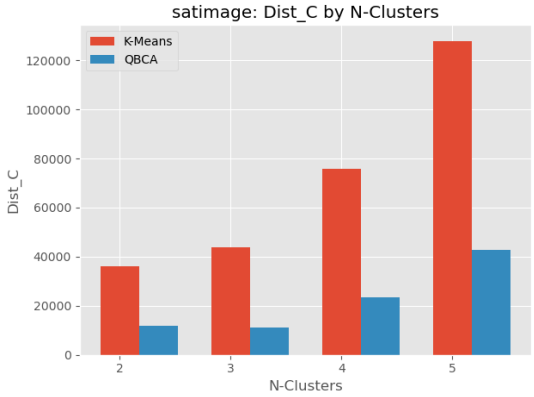

Figure 6: Distance computations on Satimage dataset with 2 dimensions.

standard principal components analysis, the plots are printed with the two most significant dimensions.

At figure 7 we can see the silhouette results obtained on this dataset, in this case we could say that K-Means slightly outperforms K-Means in the majority of instances, the only case where their performance is true is when using three centroids. The case with the most noticeable difference is the one that uses only two centroids. This could be due to the random centroid initialization of K-Means and the Gaussian based seed initialization phase of QBCA. The explanation probably is that in some cases the random clusters initialization leads to a better final distribution of the samples.

The Satimage dataset provides a very clear improvement in terms of performance, if we look at figure 6 it can be seen that a huge gap exists between both algorithms, the optimization becomes more noticeable the bigger the number of clusters is. In the case of five clusters we can see that QBCA only takes 40000 and K-Means triplicates that value with more than 120000. The increase in performance on the second dataset can be enabled by a higher number of samples and a dimensionality of fives that gives more room to exploit the quantization properties.

In terms of the qualitative performance no big differences can be observed at figure 8, as it was mentioned before the cluster ordering is what makes their colours different. The semi-triangular shape of the scattered data could be what makes it perform better with three seeds.

4.4.1 Image Segmentation

After testing on an artificial and a well known datasets some image segmentation has also been performed as the final batch of experimentation done. Two different images have been selected without any specific criteria, the images come from a well known movie scene and the recently created Cybertruck. It

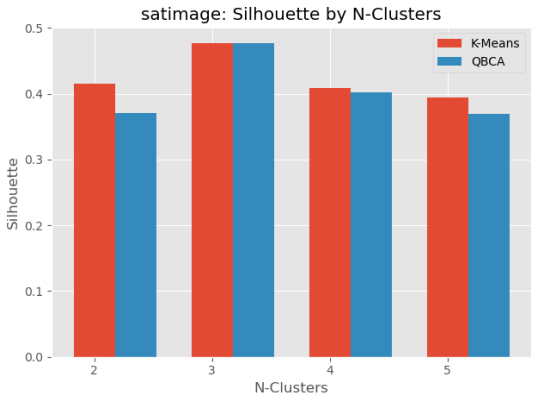

Figure 7: Silhouette scores on Satimage dataset with 2 dimensions.

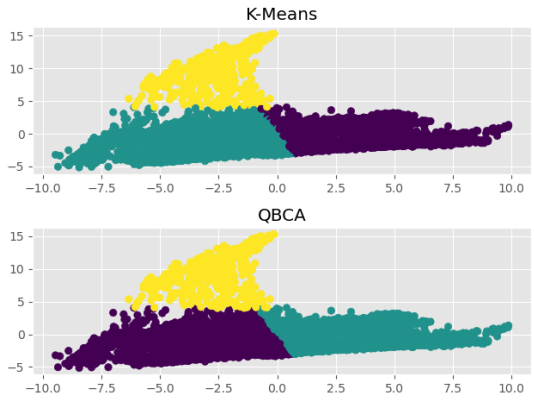

Figure 8: Clustering results on Satimage dataset with 2 dimensions and 3 seeds.

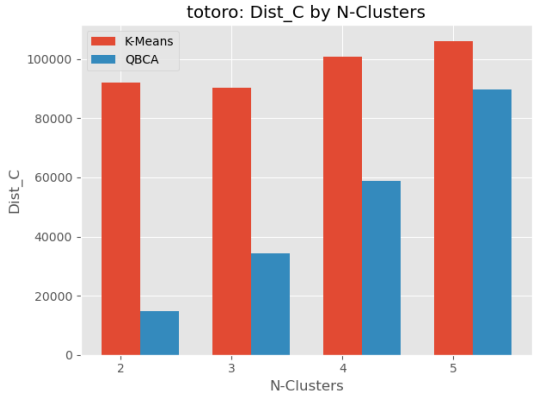

Figure 9: Distance computations on Totoro image with 5 dimensions.

is known that there are multiple techniques use to perform image segmentation with unsupervised learning, however in this case a very basic approach has been used. In this approach five attributes are used for each sample. The attributes used are the RGB values and the pixels X and Y coordinates in the image, this enables pixel position to also be relevant.

To see how well the clusters adjusted in each image for both algorithms go to figures 10 and 13. In both cases the performance is kept quite similar, however in the case of the Cybertruck a clear improvement can be seen for the configurations using four and five seeds respectively. This can be due to the consistency of the seed initialization technique used in QBCA that seems to be beneficial in this specific case, however this may not always be the case like it happened on Satimage.

In terms of the efficiency analysis two different patterns are observed in both images, in the case of Tororo (figure 9 the K-Means algorithm Starts with a way worse amount of computations that doses not substantially increase afterwards. However, the QBCA algorithm shows a different tendency in which the amounts of computations used proportionally increases with the number of clusters used. This behaviour is not the same as in the case of the Cybertruck image, if we look at figure 12 we can see that for this image K-Means becomes less efficient with more clusters and QBCA maintains its efficiency even lowering it with four seeds.

For qualitative results you can look at figures 11 and 14 where the segments are superposed over the original image, like it occurred on all the other datasets the defined clusters between both algorithms are practically the same.

4.5 Conclusions

This algorithm has been proven to noticeable lower the number of distance computations without noticeably losing cluster quality. The fact that the obtained

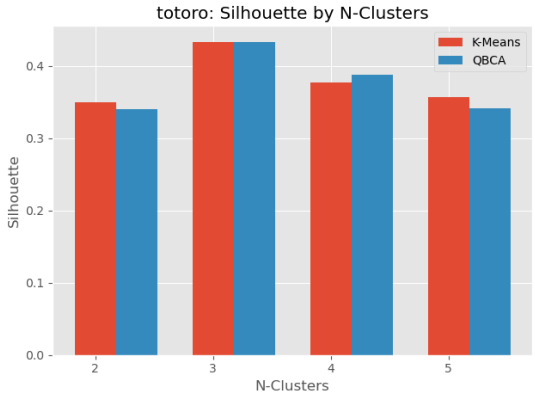

Figure 10: Silhouette scores on Totoro image with 5 dimensions.

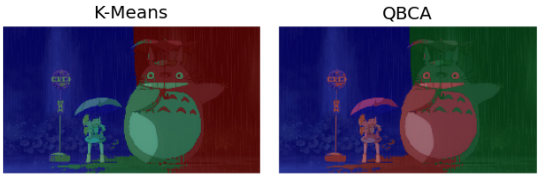

Figure 11: Segmentation results on Totoro image with 5 dimension sand 3 clusters.

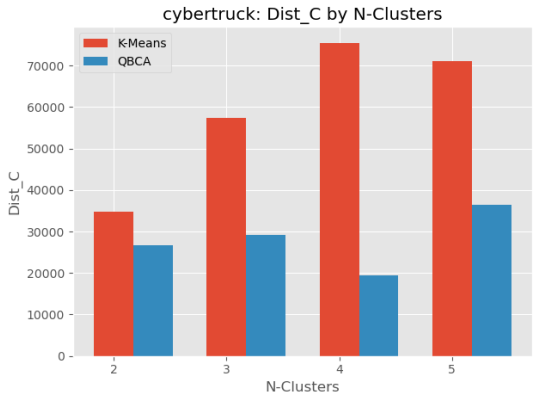

Figure 12: Distance computations on Cybertruck image with 5 dimensions.

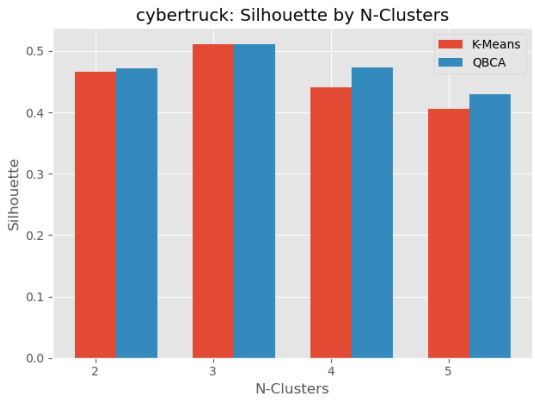

Figure 13: Silhouette scores on Cybertruck image with 5 dimensions.

Figure 14: Segmentation results on Cybertruck image with 5 dimensions and 3 clusters.

clusters are practically the same was to be expected given their similar nature, however the added consistency of using the cluster initialization in QBCA can make the clusters a bit worse in some cases. The computation times of the Sklearn implementation is better despite the optimizations, despite that the Numpy implementation used for QBCA still performs at a very fast and competent phase.

The algorithm was very clear and well explained in the original paper, in the initial phases it was a bit hard to grasp the whole concept and some more information regarding the multidimensional histogram could have been useful. Despite that all the explanations were easy to follow and implement in a very similar manner.

This has been a very great challenge and learning experience, the concept of the algorithm proposed and the way in which it exploits the bin distances is fascinating, the Dunn index algorithm [6] used in the original paper was also implemented but it has not been used because it took a way longer computation time than the algorithm itself for some reason.

4.6 Future Work

Some future work could be done with this project, the first thing could be further standardize it as a library. It could also be further optimized with the help of a code profiler like the one that comes in the Pycharm IDE.

Some improvements like the shrinking process proposed in the paper could also be included as an option to the algorithm. Finally categorical attributes could also be included as a section of the algorithm like in K-Modes.

References

[1] Raw data of the synthetic dataset used. [online] Available at:
https://github.com/deric/clustering-benchmark

[2] Satimage Dataset. [online] Available at:
https://datahub.io/machine-learning/satimage

[3] Quantization-based clustering algorithm
Zhiwen Yu, Hau-San Wong, 2010

[4] Implementation of kmeans used. [online] Available at:
https://scikit-learn.org/stable/modules/generated/Sklearn.cluster.KMeans.html

[5] Implementation of the silhouette metric used. [online] Available at:
https://scikit-learn.org/stable/modules/generated/Sklearn.metrics.silhouette_score.html

[6] Dunn index information. [online] Available at:
https://en.wikipedia.org/wiki/Dunn_index