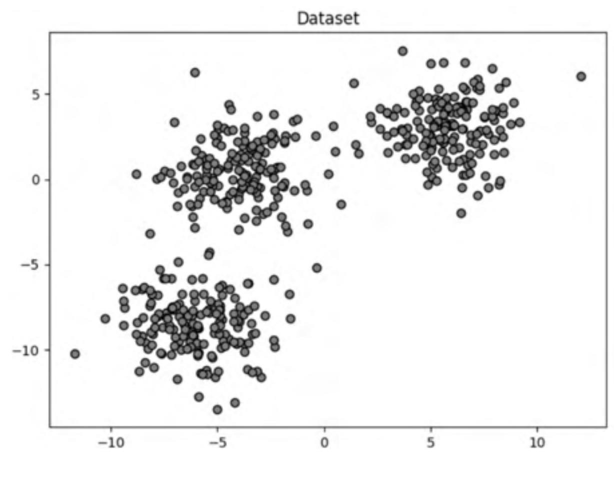
CMPE 481 Data Analysis and Visualization
FALL 2021 Assignment 1: k-Means Clustering
Zuhal Didem Aytaç – 2018400045

Table of Contents

---

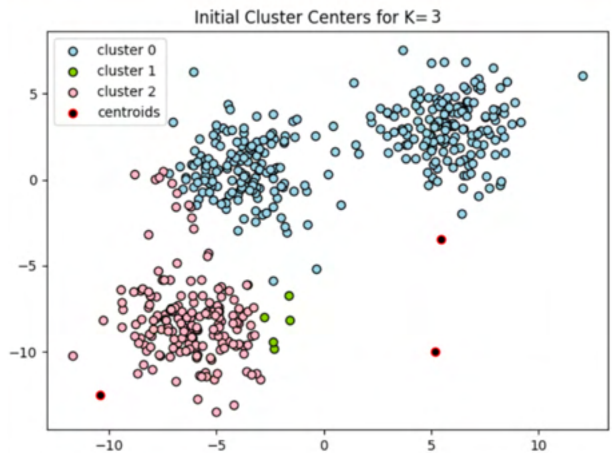
1. Generating Dataset

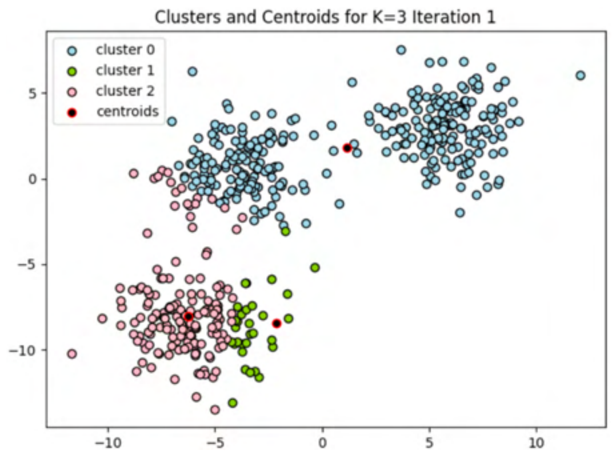
I used make_blobs function from sklearn.datasets to generate the dataset.
X, y = make_blobs(n_samples=500, centers=K_input, cluster_std=1.7, random_state=17)

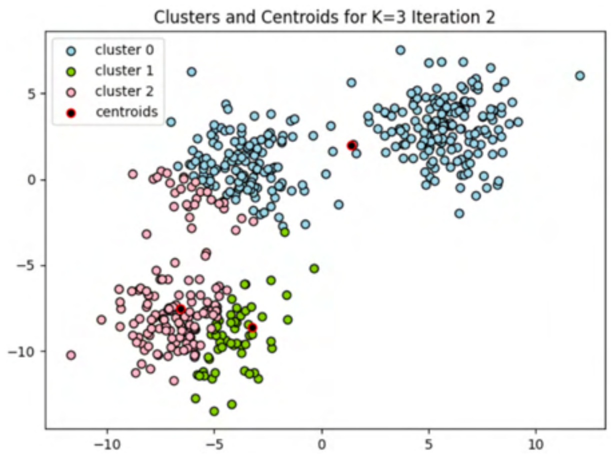


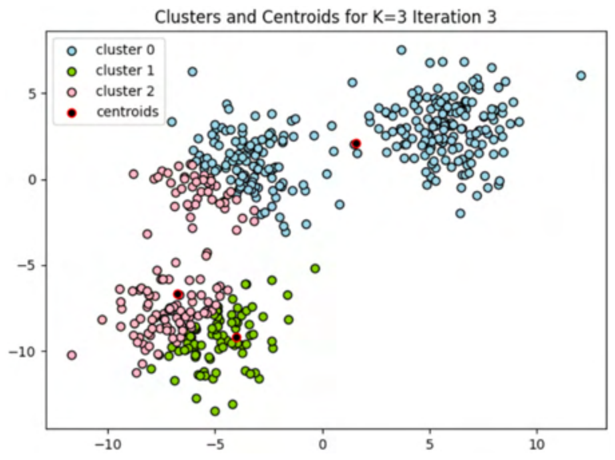
2.Applying the k-Means Algorithm

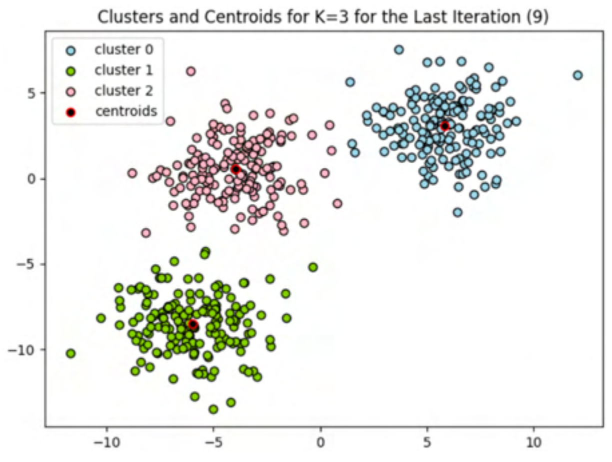
2.A. K=3



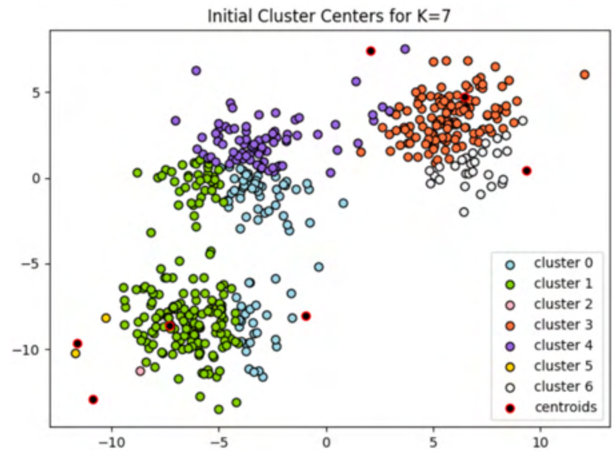


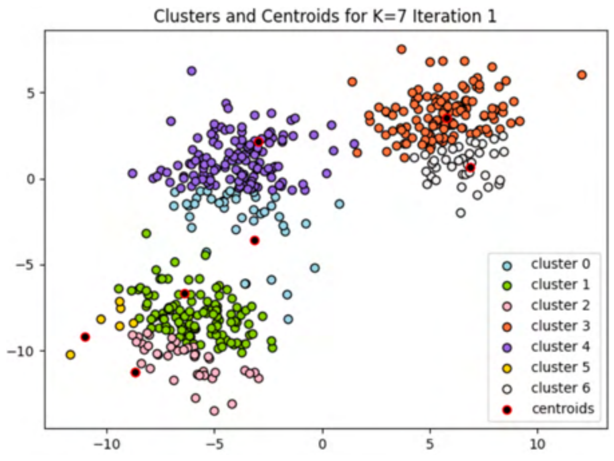


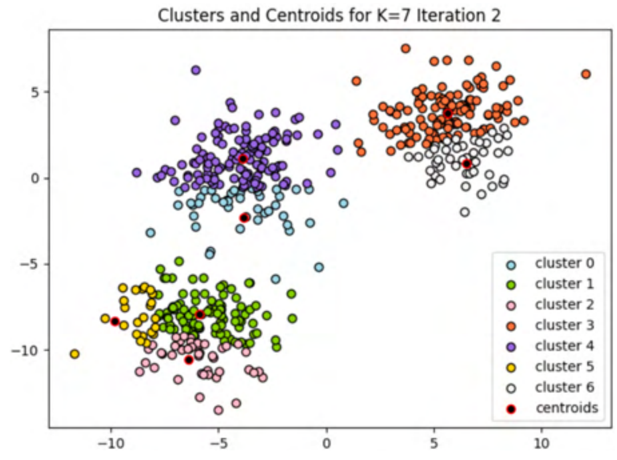


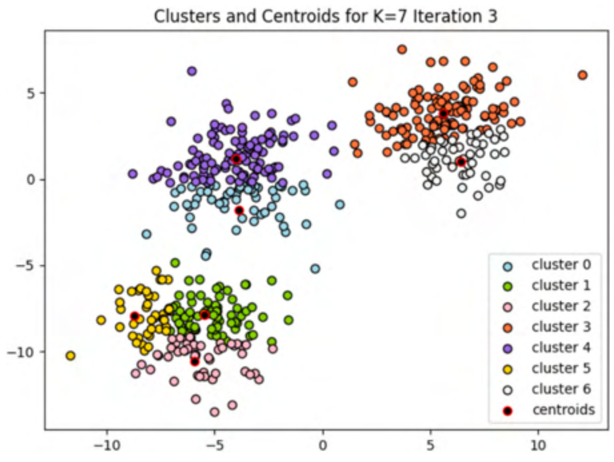


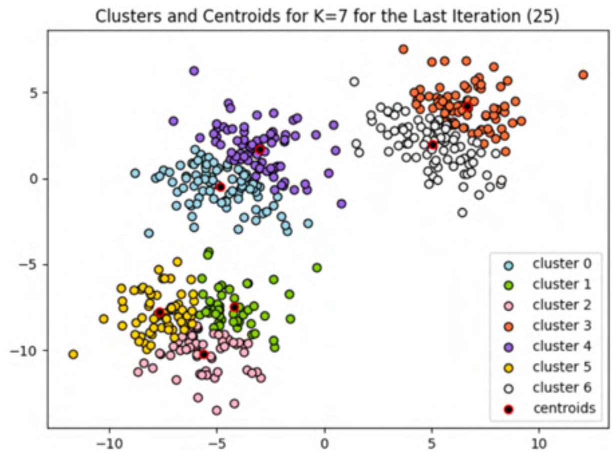
2.B. K=7


Initial Cluster Centers for K=7
cluster 0
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
centroids


Clusters and Centroids for K=7 Iteration 1
cluster 0
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
centroids

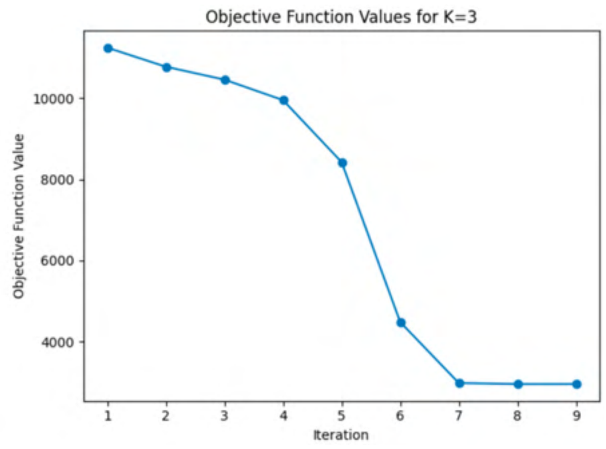

Clusters and Centroids for K=7 Iteration 2
cluster 0
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
centroids


Clusters and Centroids for K=7 Iteration 3
cluster 0
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
centroids

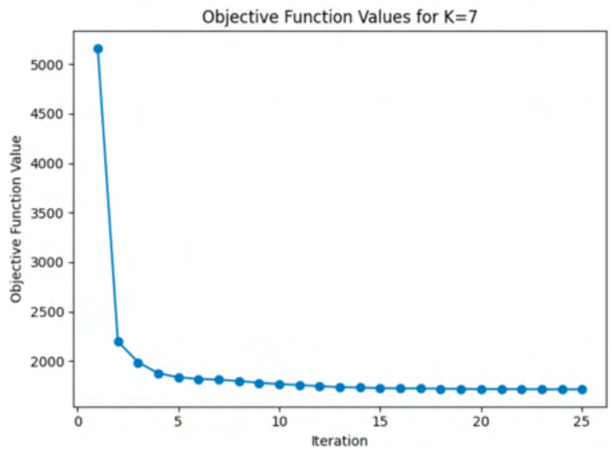

Clusters and Centroids for K=7 for the Last Iteration (25)
cluster 0
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
centroids

3. The Objective Function

3.A. K=3



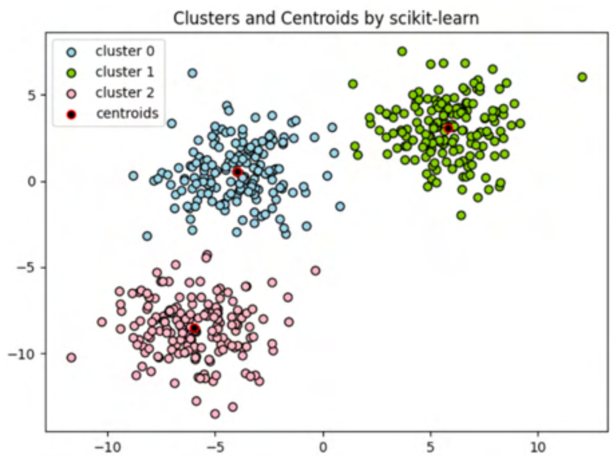
3.B. K=7

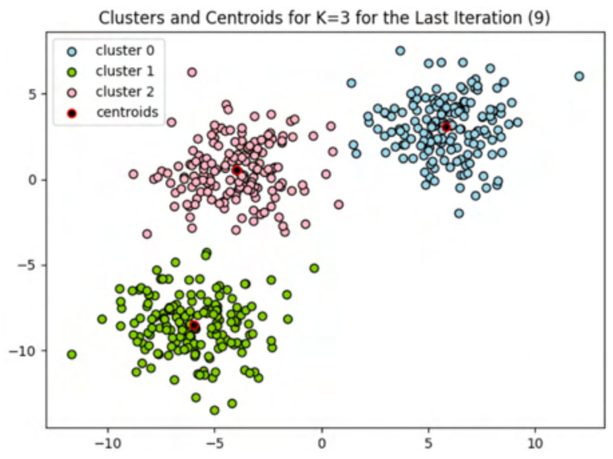


4. Comparison with scikit-learn

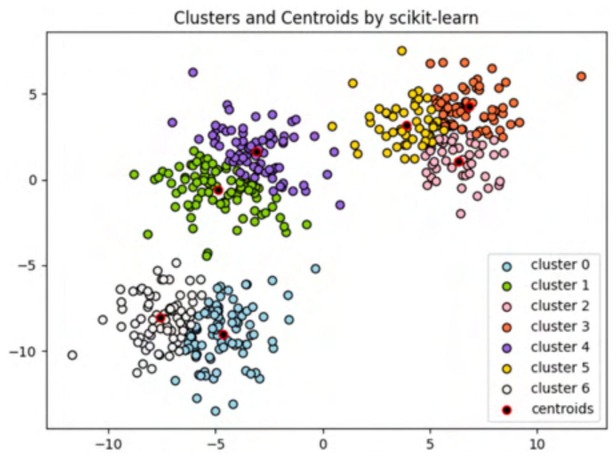
It is observed that for K=3, the cluster centers have the same position and the node-cluster assignments are the same. However, for large K, we can see that the clustering differs. This is because the algorithm tries to cluster the nodes into more groups than needed and the positions of the groups depend heavily on the initial cluster centers, which are generated randomly in my algorithm.

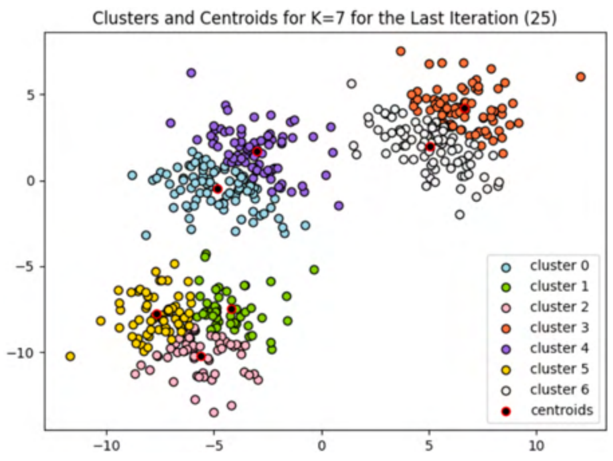
4.A. K=3





4.B K=7





5. The Best k

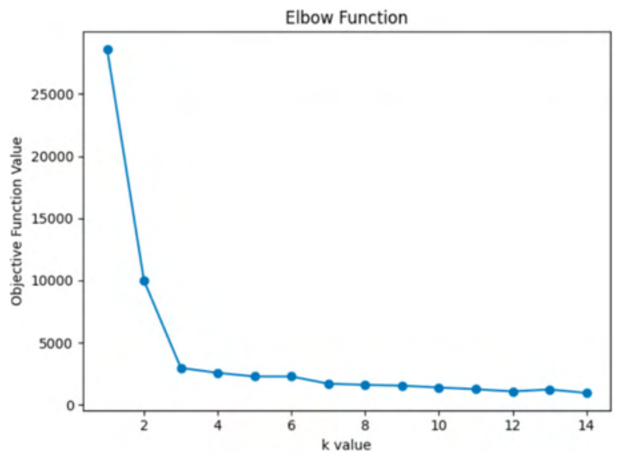
The aim of k-Means clustering method is to divide the given dataset into k clusters and whilst, minimize the within cluster variation. That value, inherently, depends on the number of clusters.

There exists plenty of different methods to find the best k for a given dataset. Of course, as k increases and at some point, equals number of data points, the within cluster variation is minimized. But that's not an intended result. I have read mostly about The Elbow Method, The Silhouette Method, Gap Statistics and Davies Bouldin Index.

To mention briefly, the Silhouette method is based on a data point's relevance with the cluster it is assigned to, compared with other possible clusters it can be assigned to. The algorithm chooses the k value where the silhouette method's output is maximized. The gap statistic method is also based on the within cluster variation value. But it compares the value *with that expected under an appropriate reference null distribution.*[1] Davies Bouldin index, indexes within cluster variation to the variation of clusters.

I chose to implement the elbow method because it's the algorithm that first came to my mind when I thought of possible methods before research. As the aim of k-Means algorithm is to minimize the within cluster variation, the elbow method compares the within cluster variation for different k values. It diminishes as k increases, because more clusters mean smaller clusters and hence, less variation within one cluster. The elbow method looks for the k value that makes a visible and significant impact on the within cluster variation value.

It is mostly easy to detect the elbow point with eye looking at the elbow function change graphs. I used the final_objective_function_values list to draw the elbow graph. This lists keeps the values of the resulting objective function value for different k values given a dataset. The graph can be seen below.



It is visible that from 1 to 2 and from 2 to 3, there are big improvements in the objective function value. However, the improvements after k=3 are ignorable and do not bring much improvement.

In python, I identified the elbow point as the point that has the maximum ratio:
(final_objective_function_values[i - 1] - final_objective_function_values[i]) / final_objective_function_values[i]

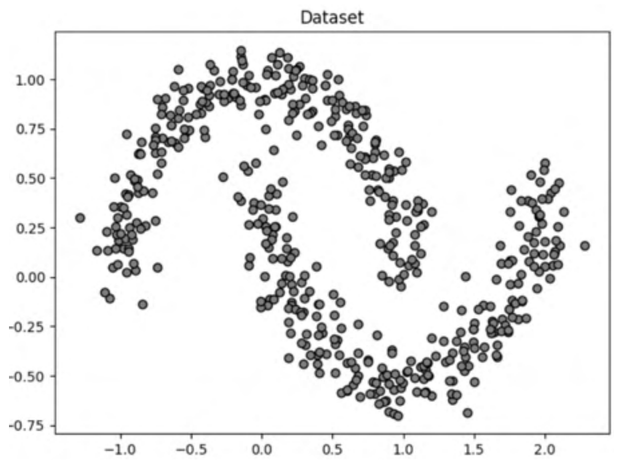
[1] https://web.stanford.edu/~hastie/Papers/gap.pdf

As the final_objective_function_values list's 0th index means k=1, the elbow point is the point with highest ratio + 1. This algorithm succeded to find best k as 3 with the given dataset. It also succeeded for different datasets with different number of cluster centers.

6. Non-Convex Dataset

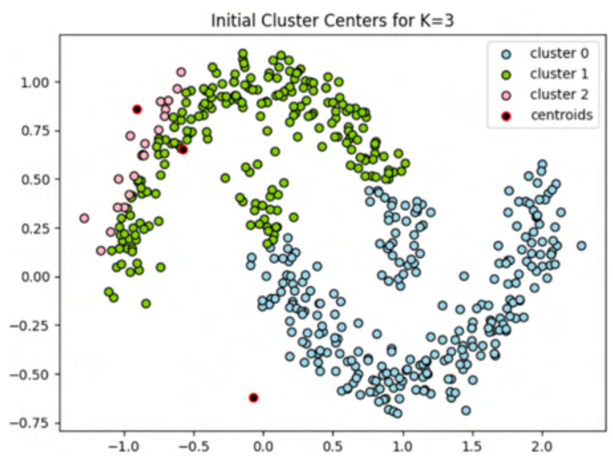
6. 1. Generating Dataset

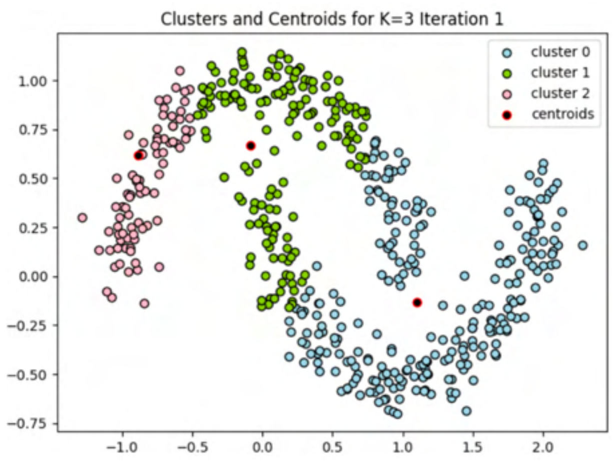
I used make_moons function from sklearn.datasets to generate the dataset.
X, y = make_moons(n_samples=500, noise=0.1, random_state=17)

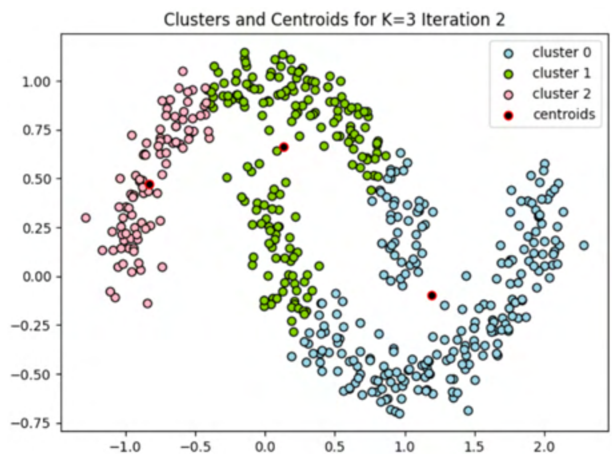


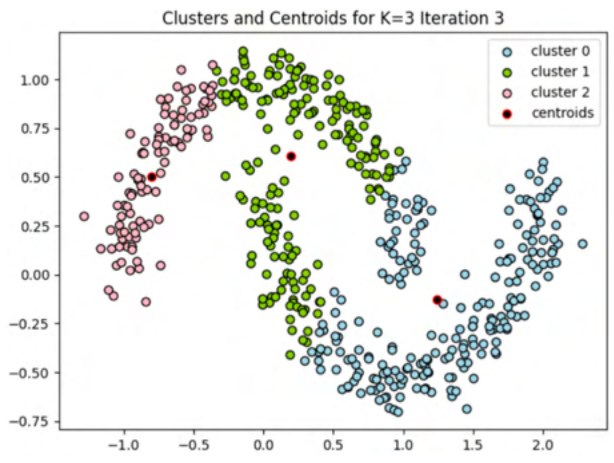
6.2.Applying the k-Means Algorithm

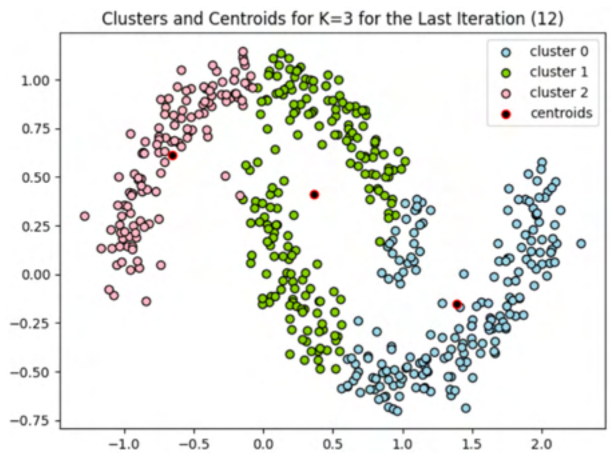
6.2.A. K=3



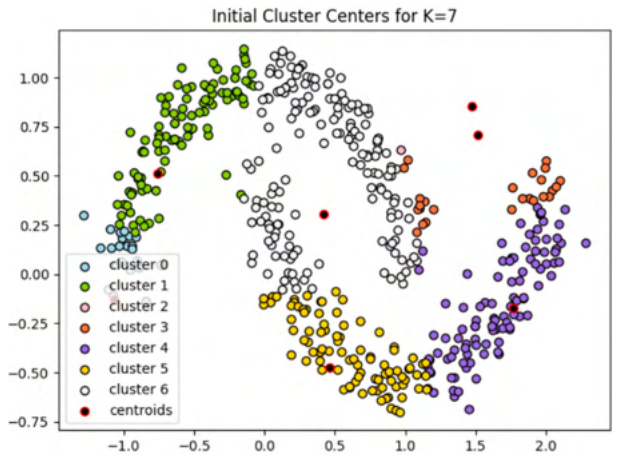


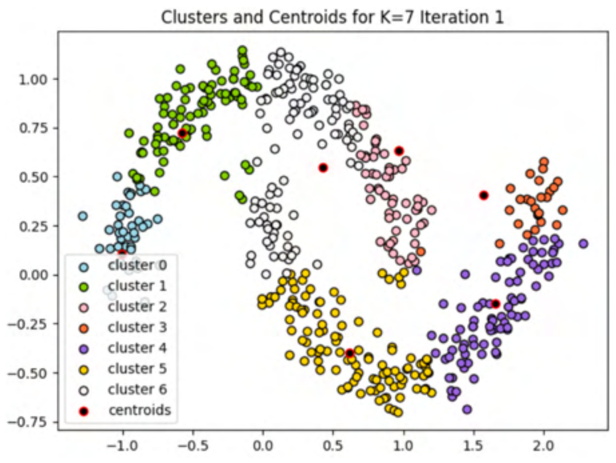


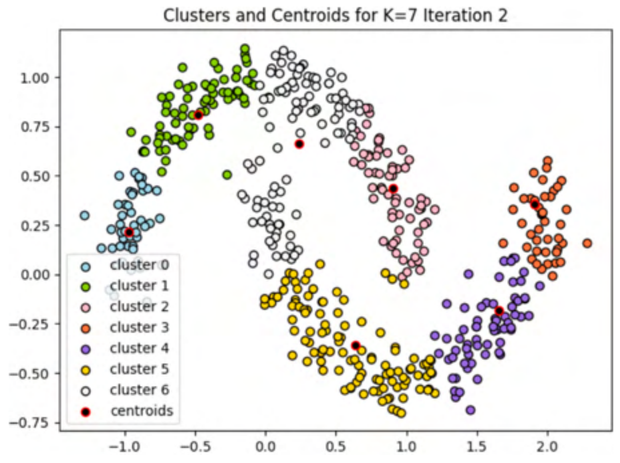


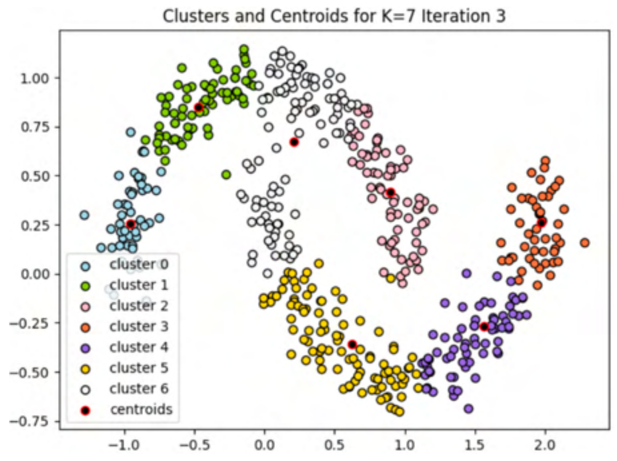


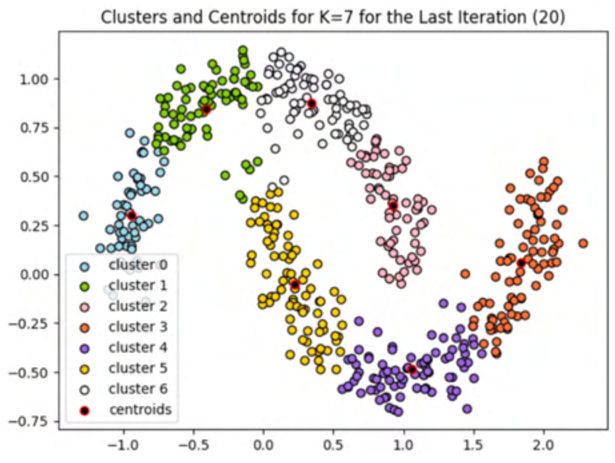
6.2.B. K=7


Initial Cluster Centers for K=7
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
cluster 0
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
centroids


Clusters and Centroids for K=7 Iteration 1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
cluster 0
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
centroids

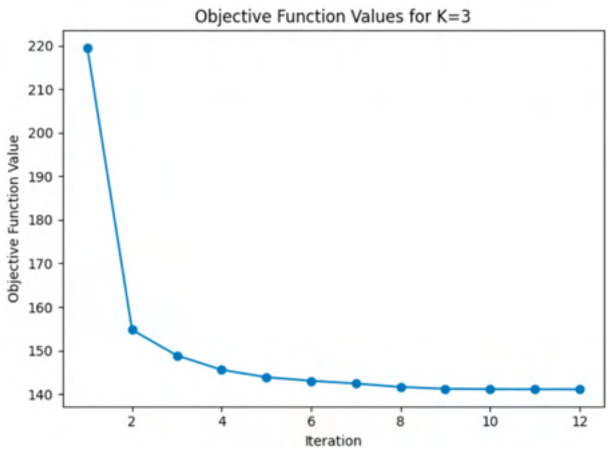

Clusters and Centroids for K=7 Iteration 2
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
cluster 0
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
centroids


Clusters and Centroids for K=7 Iteration 3
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
cluster 0
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
centroids

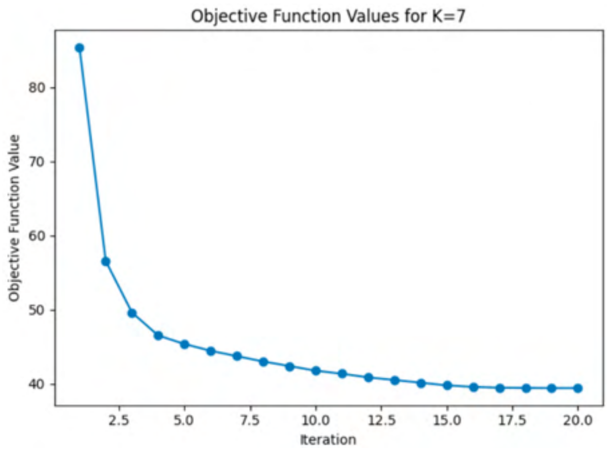

Clusters and Centroids for K=7 for the Last Iteration (20)
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
cluster 0
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
centroids

6.3. The Objective Function

6.3.A. K=3



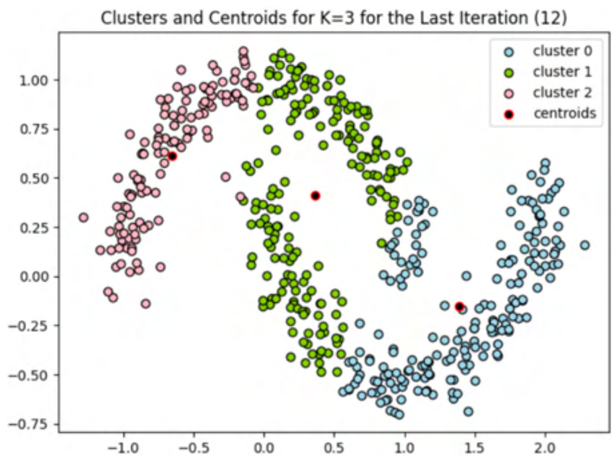
6.3.B. K=7

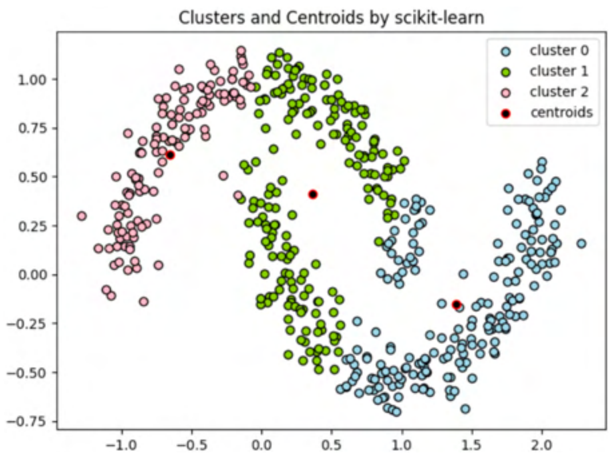


6.4. Comparison with scikit-learn

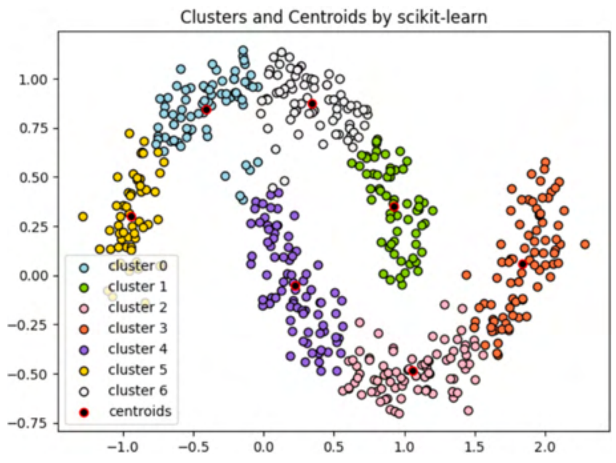
Although the results are unsuccessful, they are still very similar to skicit-learn. This is because k-means is not suitable for non-convex datasets.

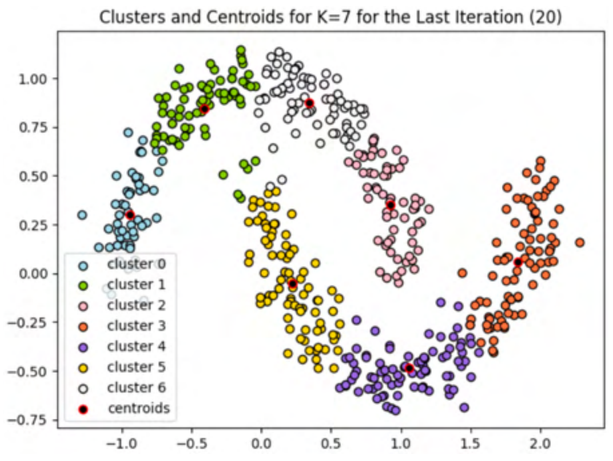
6.4.A. K=3





6.4.B K=7


Clusters and Centroids by scikit-learn
cluster 0
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
centroids


Clusters and Centroids for K=7 for the Last Iteration (20)
cluster 0
cluster 1
cluster 2
cluster 3
cluster 4
cluster 5
cluster 6
centroids