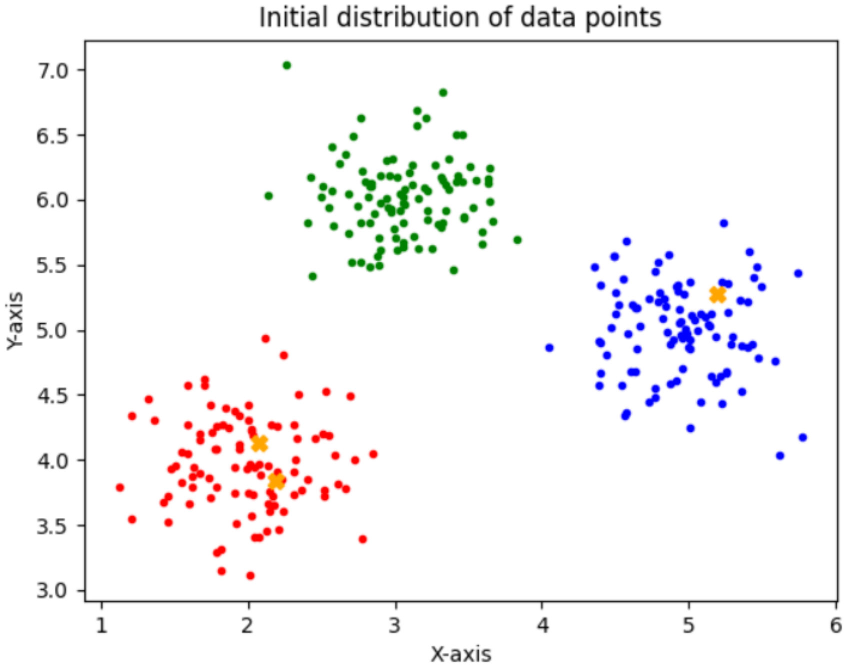
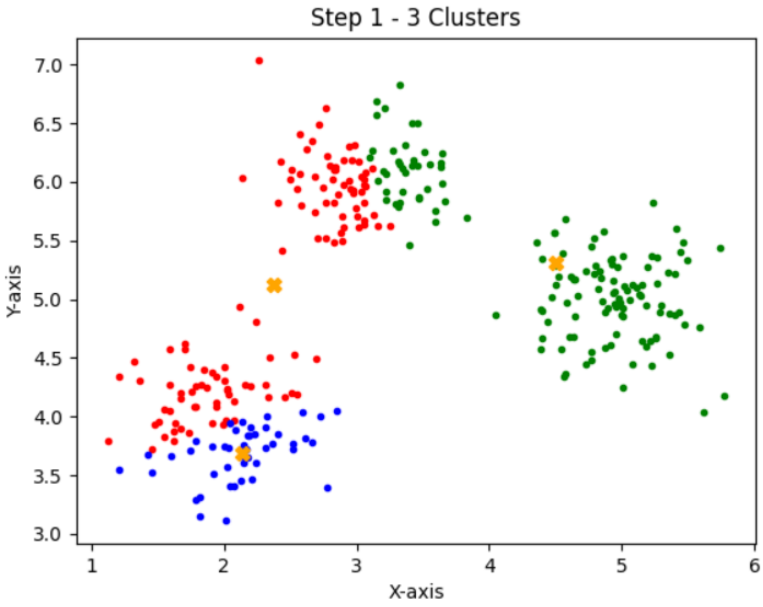
Alper Canberk Balcı - 2017400087
21/11/2021

CMPE481 Data Analysis and Visualization Project-1

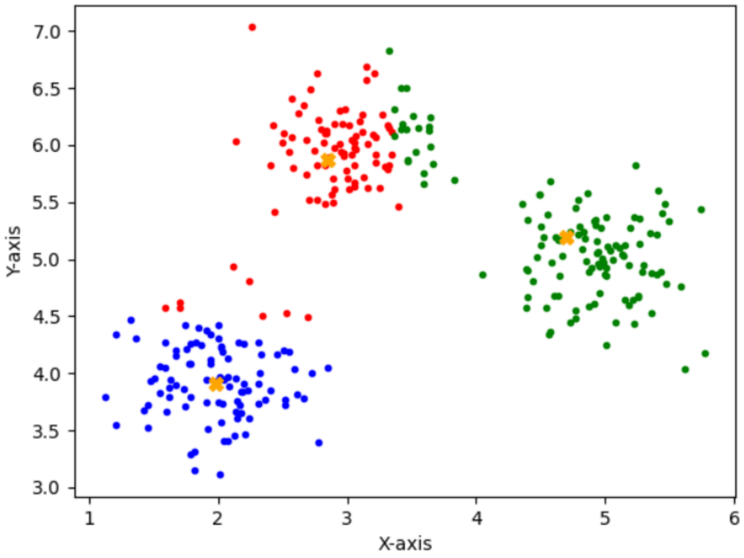
1) I generated a 2D toy/simple dataset suitable for clustering, k=3.
   The function is: generate_clusters_blobs(centers, cluster_std)
   My code has helping comments, too.

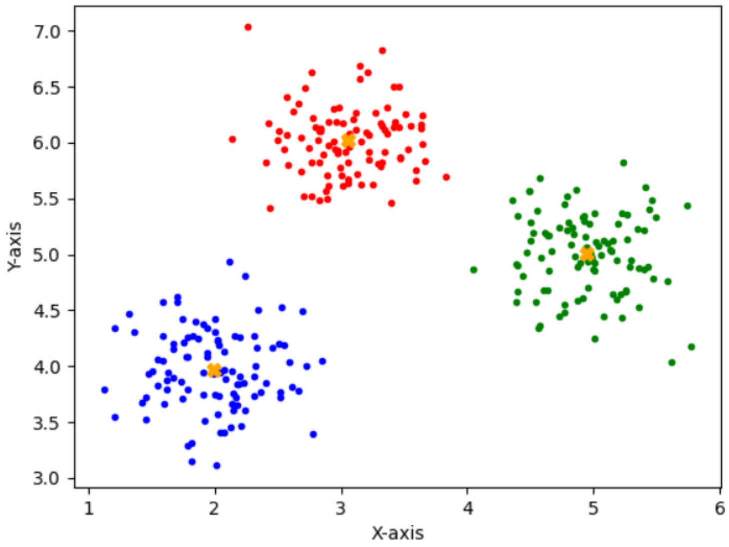


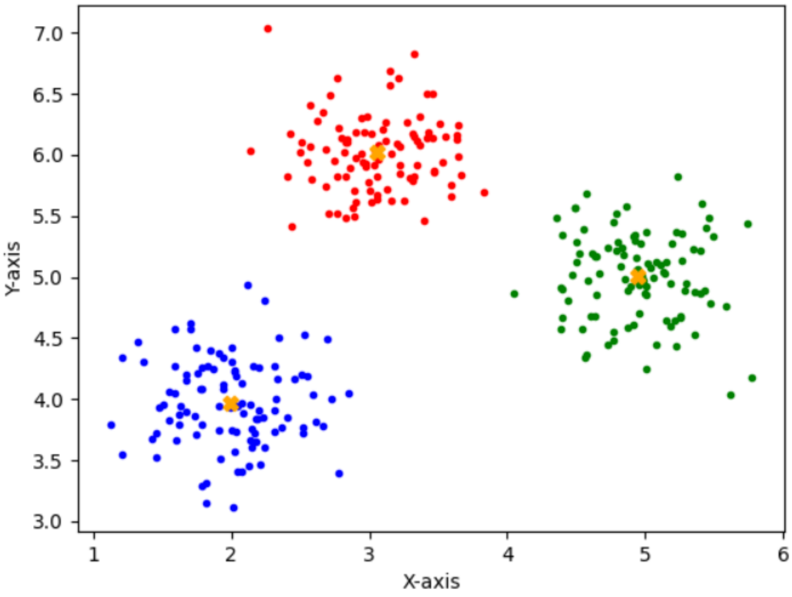
2) My algorithm works correctly. It shows the initial random cluster centers, and then; for the first three iterations and the last iteration, the clusters and cluster centers. It also shows, at the last iteration, the change in the objective function value for the steps as a graph/plot. I use the function in a for loop in the main function, to use Elbow Method later.
   But before that, I use K_means(3) and K_means(5) as it is expected to be used with two different cluster numbers in the project description. K = 3 images are below. K = 5 images are at the end of the report.



Alper Canberk Balcı - 2017400087
21/11/2021

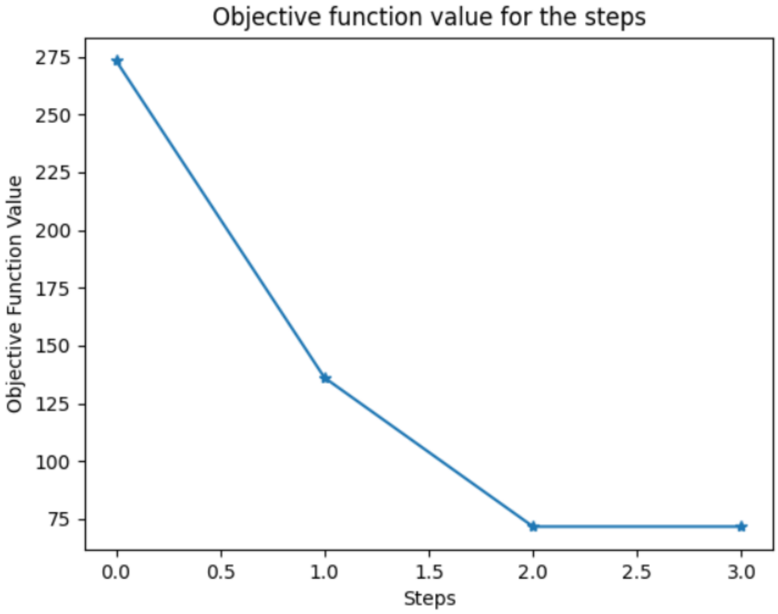

Step 2 - 3 Clusters


Step 3 - 3 Clusters

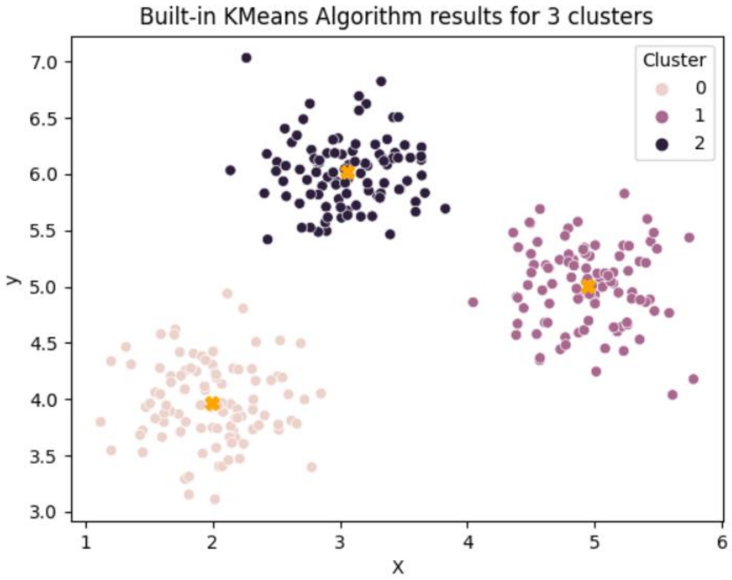

Final Step and Situation of Clusters

Alper Canberk Balcı - 2017400087
21/11/2021

3) The code has some outputs as print statements. K_means function returns some output to print them as a kind of report. It prints the cluster centers found for K clusters, the number of iterations to find them, and the final objective function value. It is in a for loop, that's why it shows for clusters K = [2,8]
For the step 3 in the project description, it is working it shows objective function vs iteration count for all iterations



4) I used scikit's k-Means algorithm and plotted final cluster centers. It's pretty much the same result.

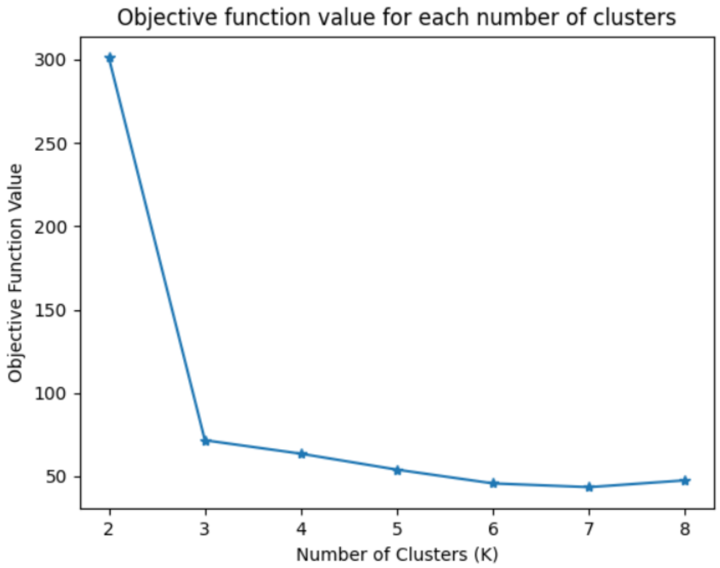


Alper Canberk Balcı - 2017400087
21/11/2021

5) Implementing a method to find best k automatically.

To perform k-Means clustering, we must decide how many clusters we expect in the data. The problem of selecting k is not very simple. Usually, it is given by the user. The best version of a run of k-Means algorithm for a number of cluster k is the one with the smallest value of within-cluster variation, that is the objective function value.

My algorithm already plot this for each of the cluster count options from 2 to 8. Of course, there might be changes in different runs for a cluster count, because k-Means is a local search procedure, and the final cluster centers highly depend on the initial cluster centers.



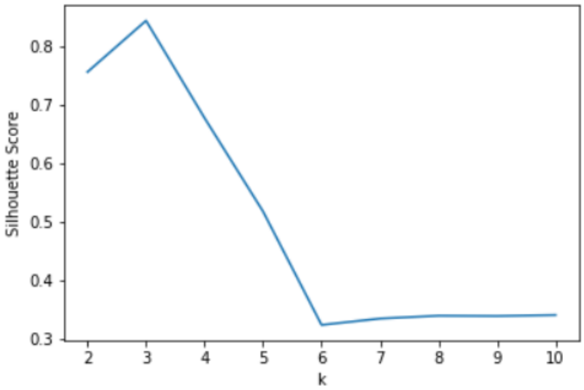
There is a method to find best k for the algorithm, called Elbow Method. It takes advantage of the objective function value for each number of clusters data. Sum of squared distances is the actual value, and it tends to go to zero as number of clusters increase. However, we want an efficient and correctly working algorithm to find best number of clusters, k.

This graph looks like an arm, that has an elbow on it. The elbow point is the optimal value for k. That is the main idea of the Elbow Method. The reason behind this idea is that from adding more clusters to the algorithm, we do not get much information out of it, not to mention, finding wrong number of, an excessive one, clusters. So, minimizing objective function value is not the thing we are looking for, as it means just one cluster for each observation, and it is not practical nor correct.

There is also the Silhouette Method. The silhouette value measures how similar a point is to its own cluster (cohesion) compared to other clusters (separation).[1]
The range of the Silhouette value is between +1 and -1. A high value is desirable and

[1] https://en.wikipedia.org/wiki/Silhouette_(clustering)

indicates that the point is placed in the correct cluster. If many points have a negative Silhouette value, it may indicate that we have created too many or too few clusters.[2]



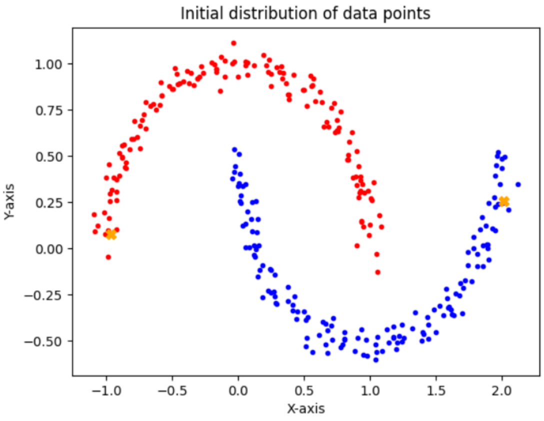
It looks like this(above) for the given dataset in the source. The Elbow Method is more of a decision rule, while the Silhouette is a metric used for validation while clustering. Thus, it can be used in combination with the Elbow Method.

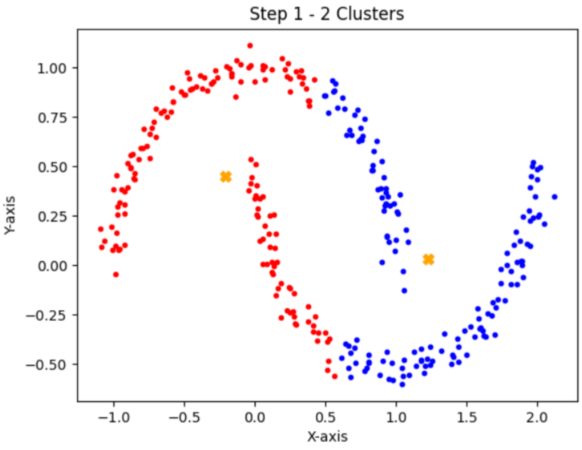
Therefore, the Elbow Method and the Silhouette Method are not alternatives to each other for finding the optimal K. Rather they are tools to be used together for a more confident decision.

I will not implement this part as it uses the sci-kit learn library and writing it from scratch is not feasible for this project.

I showed 3 cluster results above, I will not show it here.

Doing the experiment with my code with a strange dataset

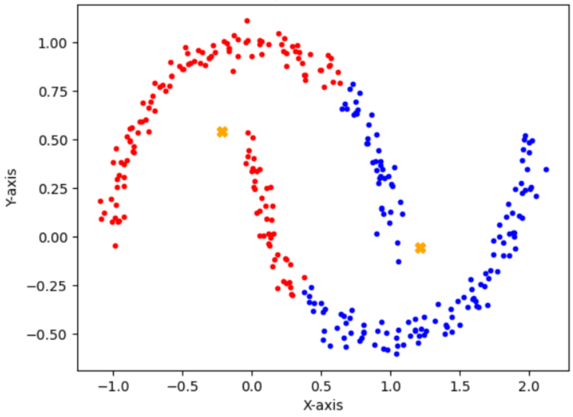




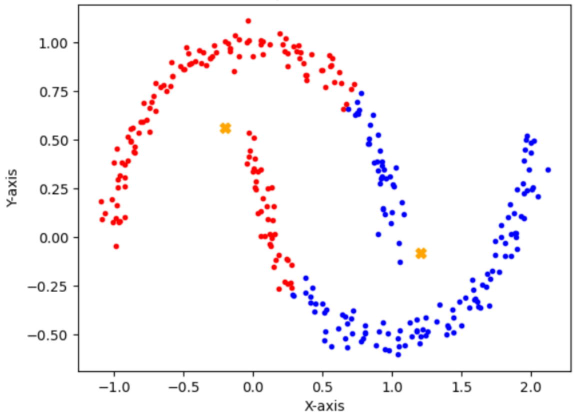
[2] https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb

Alper Canberk Balcı - 2017400087
21/11/2021

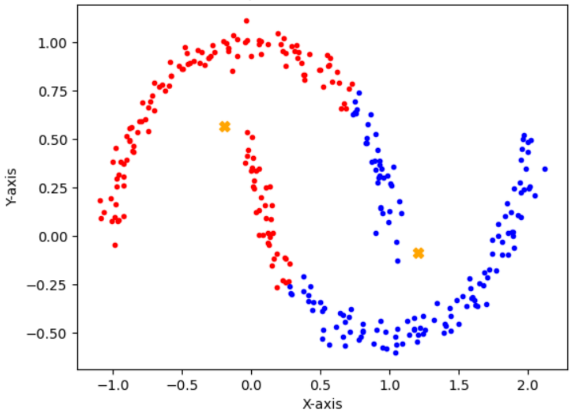
Step 2 - 2 Clusters



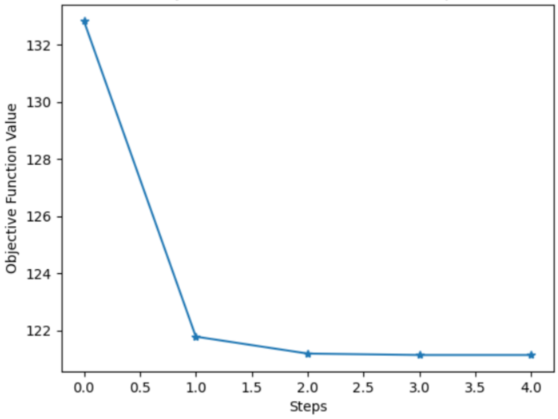
Step 3 - 2 Clusters



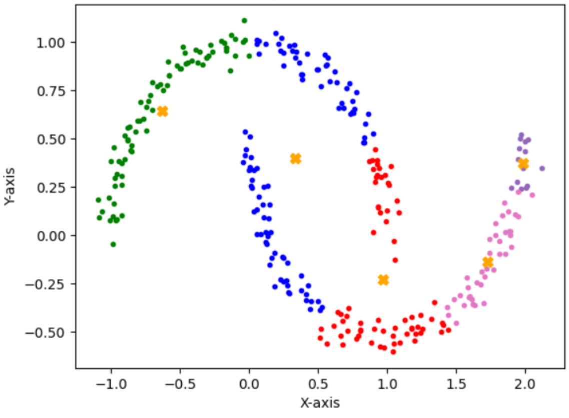
Final Step and Situation of Clusters



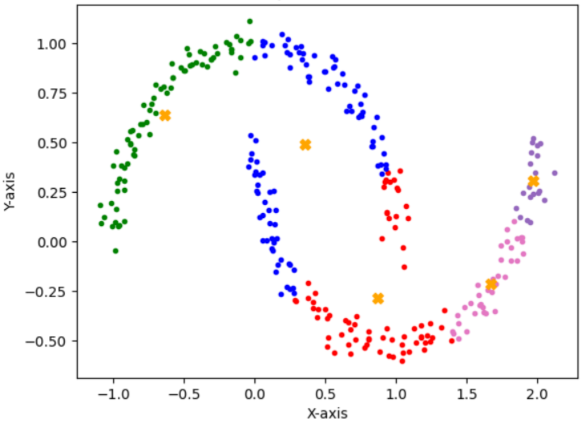
Objective function value for the steps



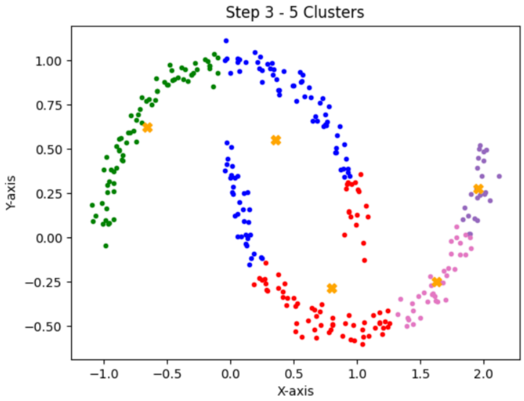
Step 1 - 5 Clusters

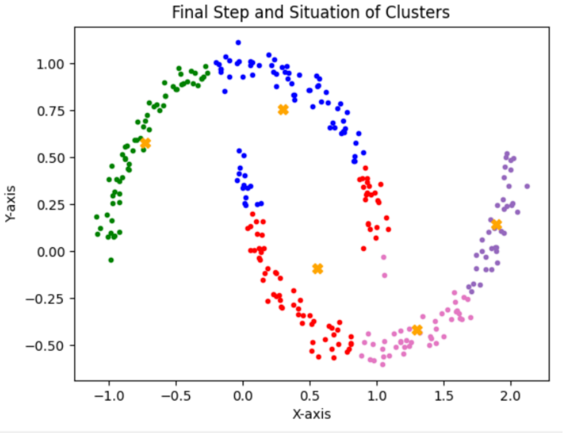


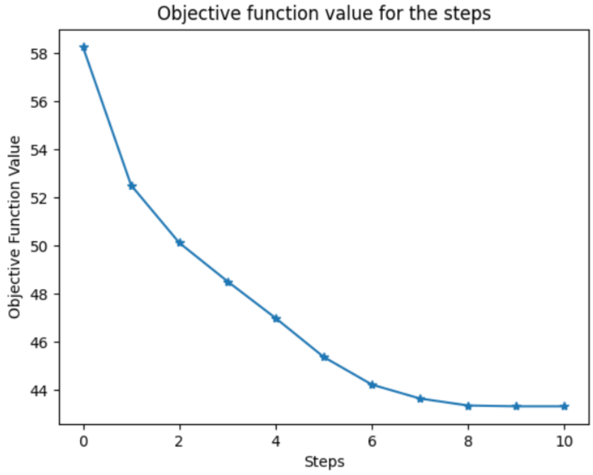
Step 2 - 5 Clusters

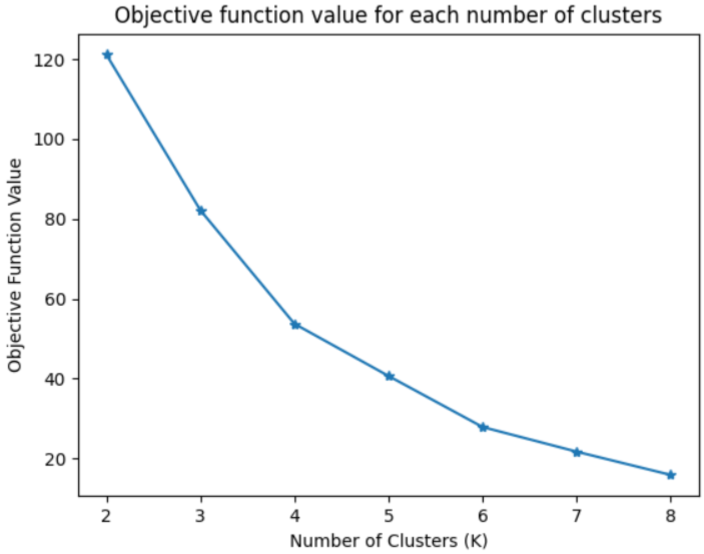


Alper Canberk Balcı - 2017400087
21/11/2021



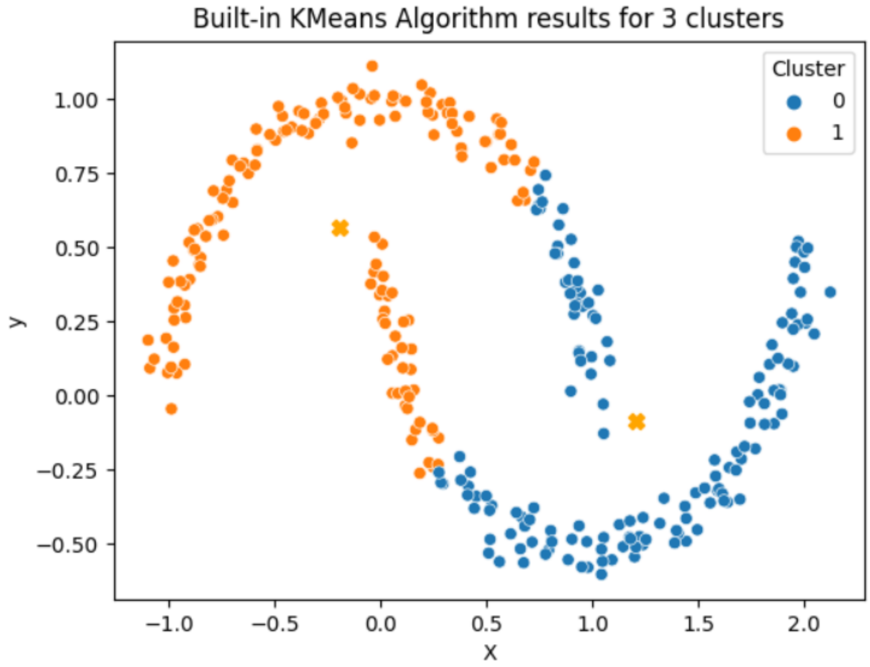






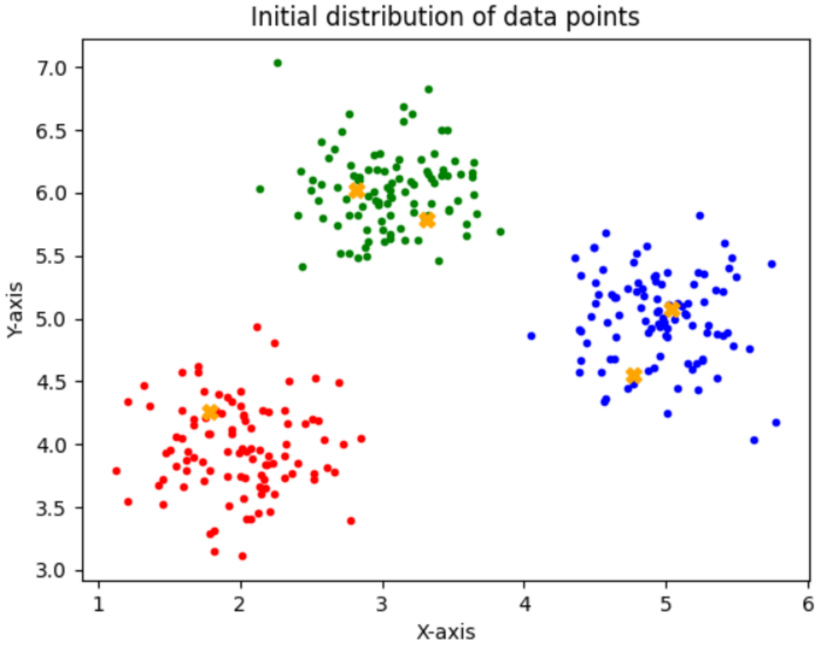
And at last, the built-in k-means

Alper Canberk Balcı - 2017400087
21/11/2021

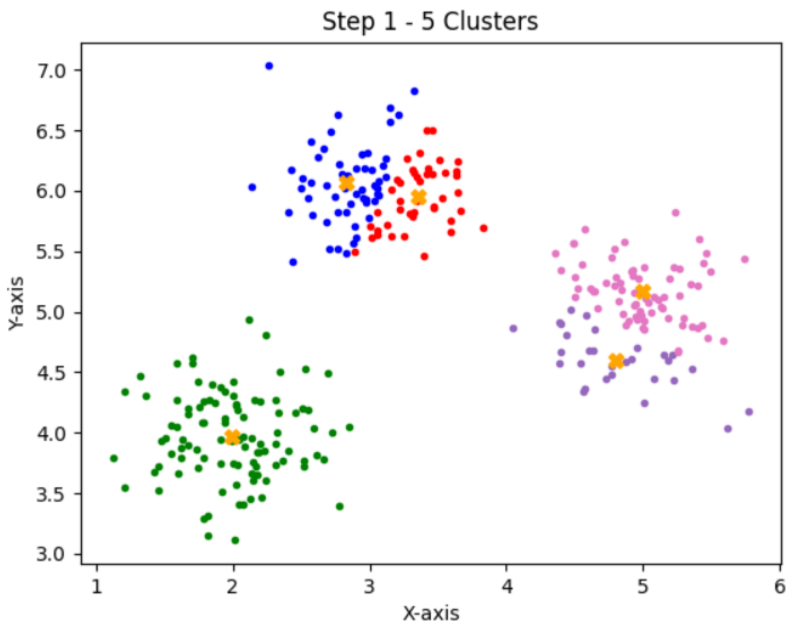


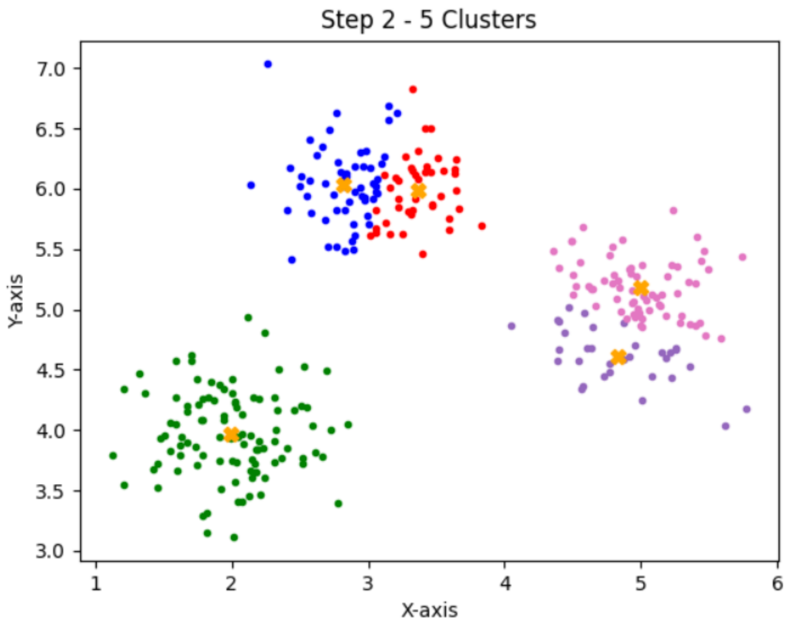
Extra Images

Step-2, K=5 images

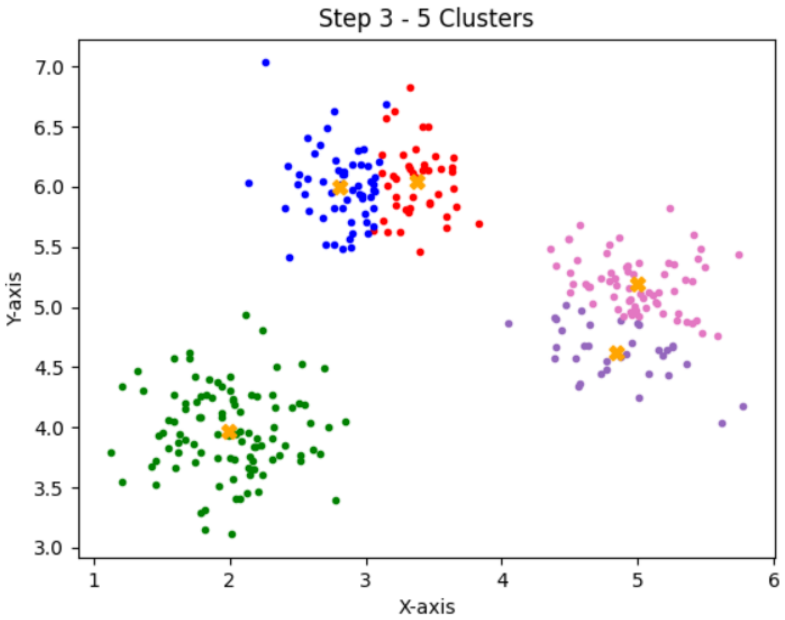


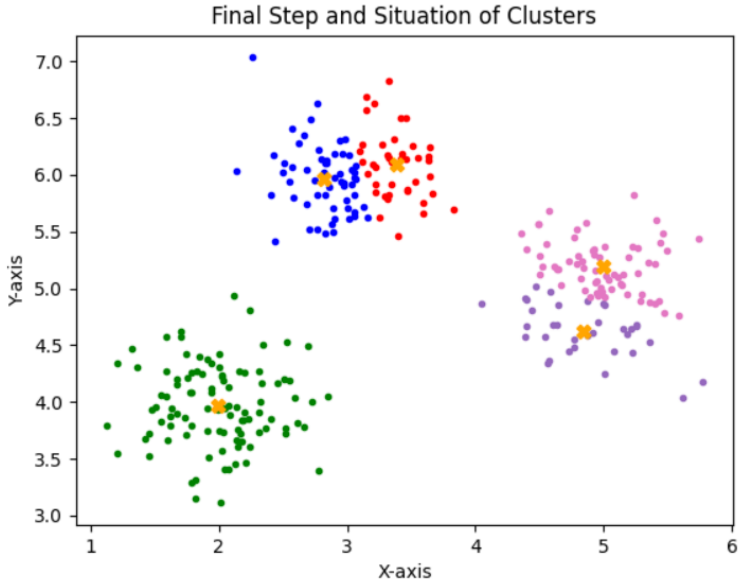
Alper Canberk Balcı - 2017400087
21/11/2021

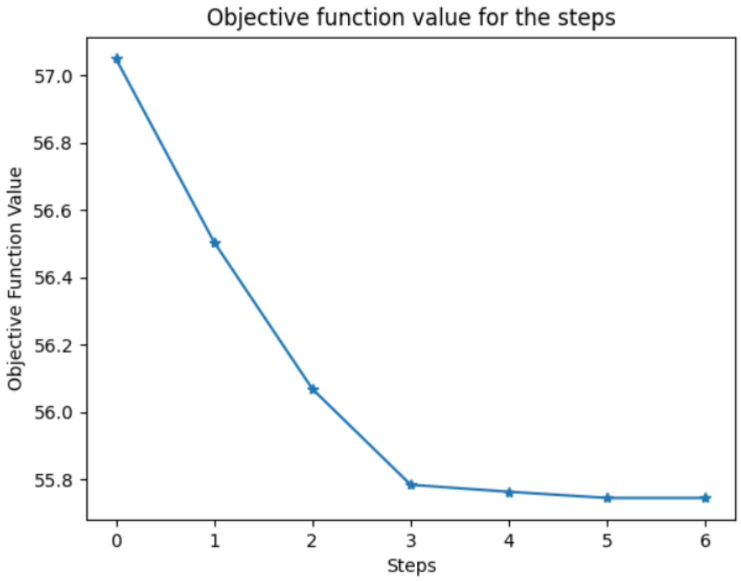




Alper Canberk Balcı - 2017400087
21/11/2021







Resources

https://en.wikipedia.org/wiki/Silhouette_(clustering)

https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb