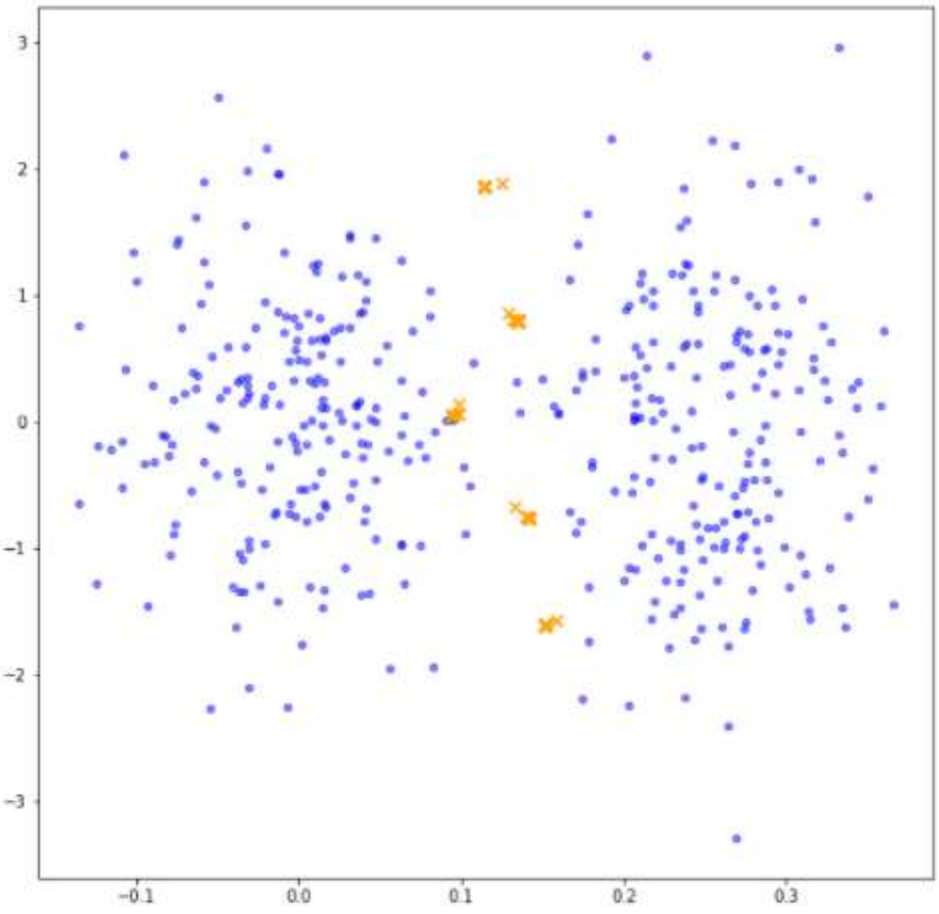
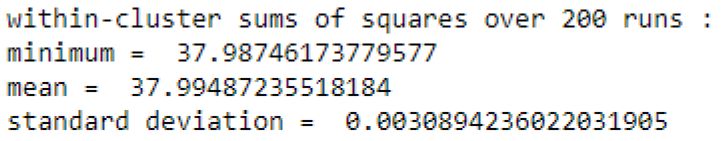
Attached file: Clustering.ipynb

In order to have a better understanding of how to divide the data, I draw the result of the last clustering performed in the loop in sections a, b, e, f on the graph with the similar data center at the end of the description of these sections and in c, d I also make a clustering with the numbers from the graph and draw it. The code part of the plot_clusters function in the first work is used at the end of these parts and I will give its result here at the end of each part.

a) The file first_clustering_dataset.csv is drawn in the form of a distribution diagram of its first and second column characteristics. The points are drawn as transparent blue circles to better show the points of data concentration and overlapping data. With sklearn.cluster.KMeans, which is the implementation of the same Lloyd's algorithm, I divided the data into 5 clusters 200 times in a row with the kmeans method by choosing the center of the initial random clusters, and each time the obtained 5 centers were shown on the graph as I drew a transparent orange cross. These centers were found next to each other every time, so that they are not transparent on the other figure. For each of these 5 solution centers, 2 focus points have been found and they are close to each other.

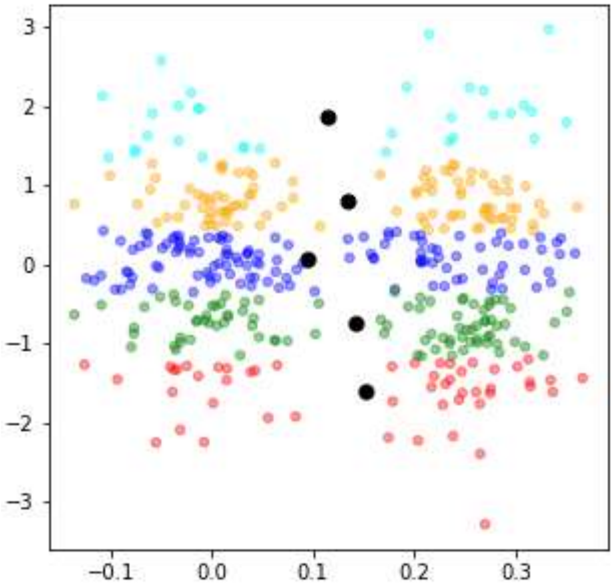


I kept the value of the cost function $J(C,L)$ obtained in the sse array at each time of clustering. Finally, I calculated the minimum value, average and standard deviation on these elements. The average error is not small, but the smallness of the standard deviation shows the concentration of the obtained centers as seen in the figure. Despite the randomness of the first choice and the sensitivity of this algorithm on these initial values, no such sensitivity was observed in this issue. The data of this problem are closer to the two clusters.

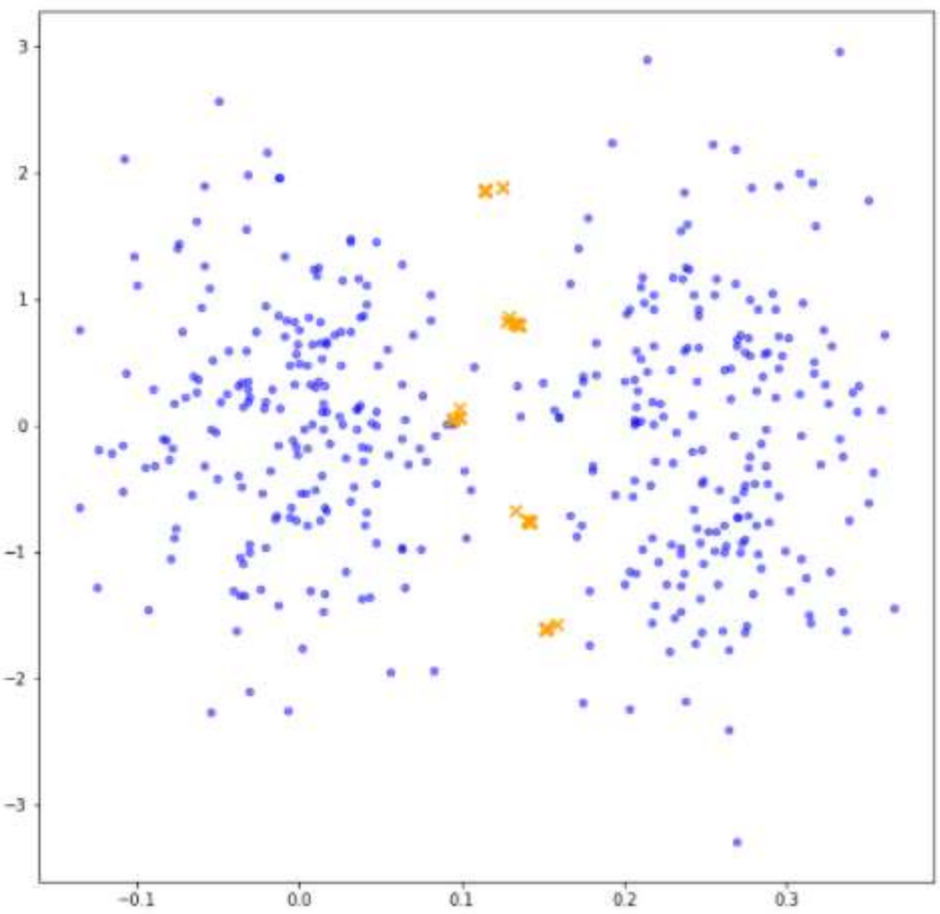


```
within-cluster sums of squares over 200 runs :
minimum =  37.98746173779577
mean =  37.99487235518184
standard deviation =  0.0030894236022031905
```

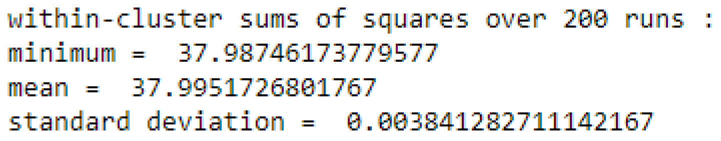
The diagram of the last clustering performed:



b) The same data as before, this time with the first value method, instead of randomly, we chose a more intelligent method of Kmeans++ method. The implementation of this section is the same as the previous section. You can see the diagram below:

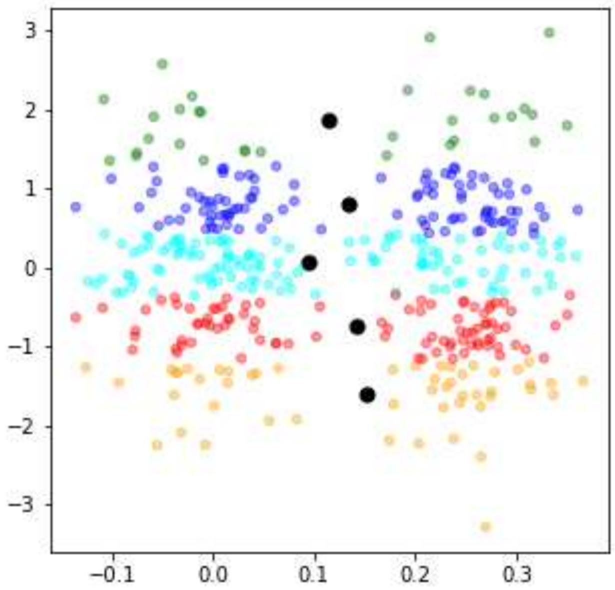


The difference between this mode and the previous mode is very small and can be ignored. On this issue, both methods can be considered equivalent.

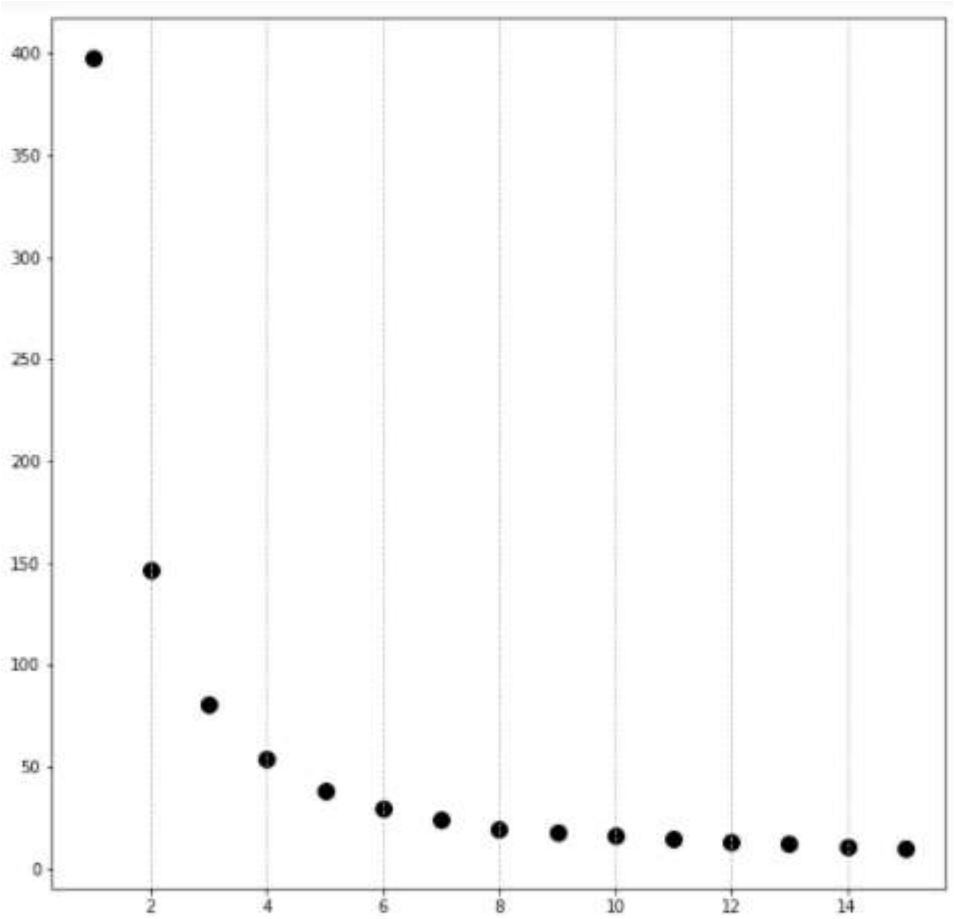


```
within-cluster sums of squares over 200 runs :
minimum =  37.98746173779577
mean =  37.9951726801767
standard deviation =  0.003841282711142167
```

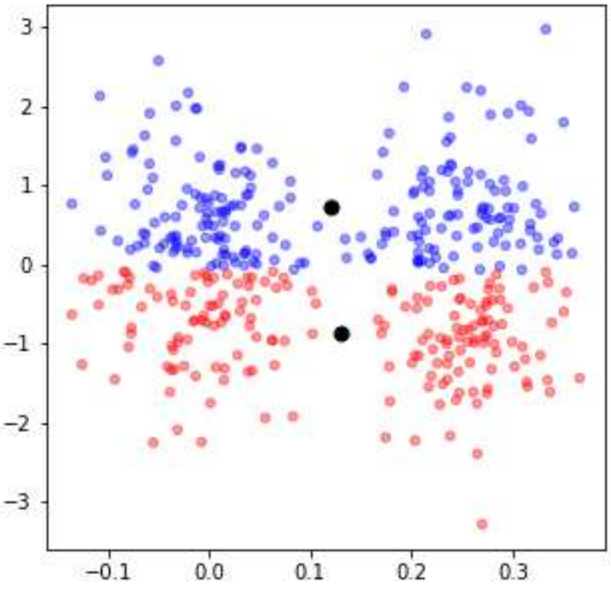
The diagram of the last clustering performed:



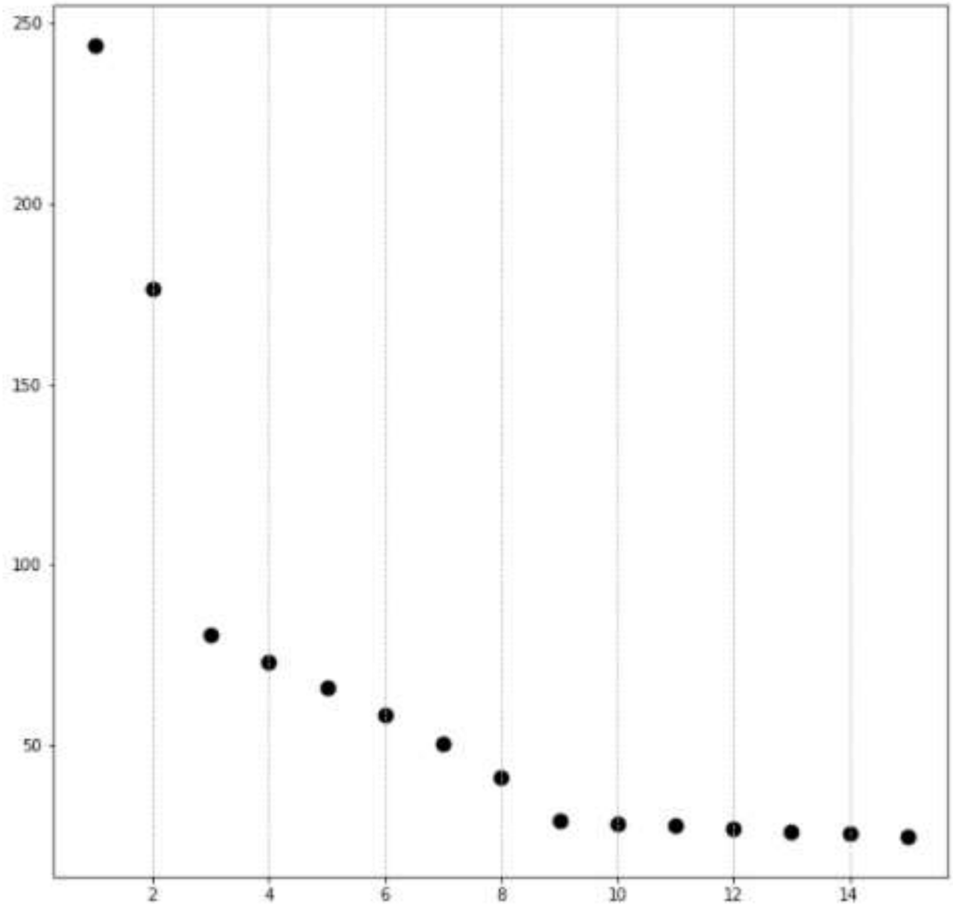
c) With the same data as before, this time the goal is to find that knee point on the graph $J(C,L)$ in terms of the number of clusters. For each k between 1 and 15, I clustered 200 times by the method of part a with a random initial center and recorded the lowest value of $J(C,L)$ obtained on the graph. For this data, this number is equal to 2. If we look at the form of the data in the previous section, this number is the first polynomial of the data of this problem that can be seen on the graph.



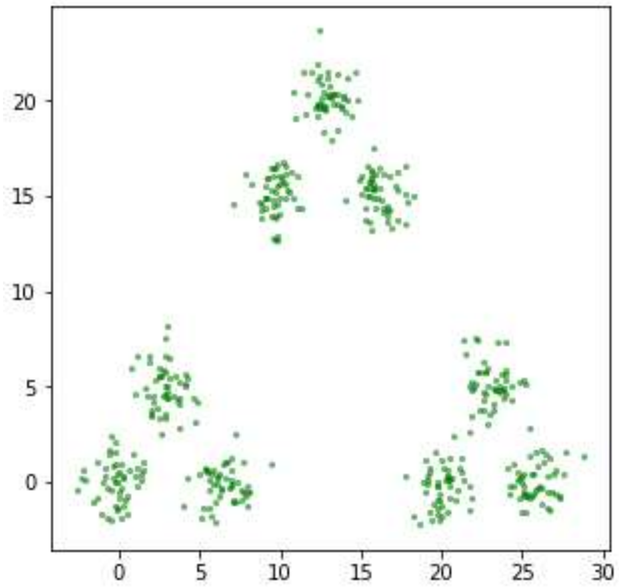
For example, a clustering with k=2:



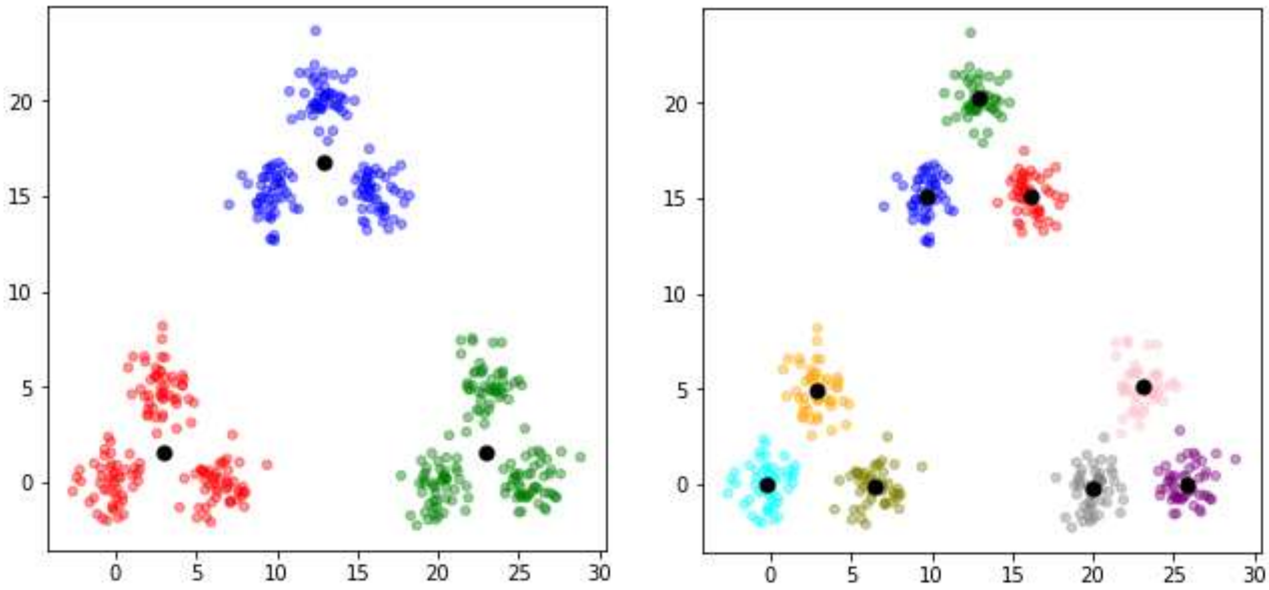
d) I repeat part c this time with the data of the second_clustering_dataset.csv file, with the difference that instead of the minimum of $J(C,L)$ I record its root value as suggested on the graph. As mentioned in the question, this graph means two points, the first one that shows a bigger difference is equal to 3 and the second one is on 9.



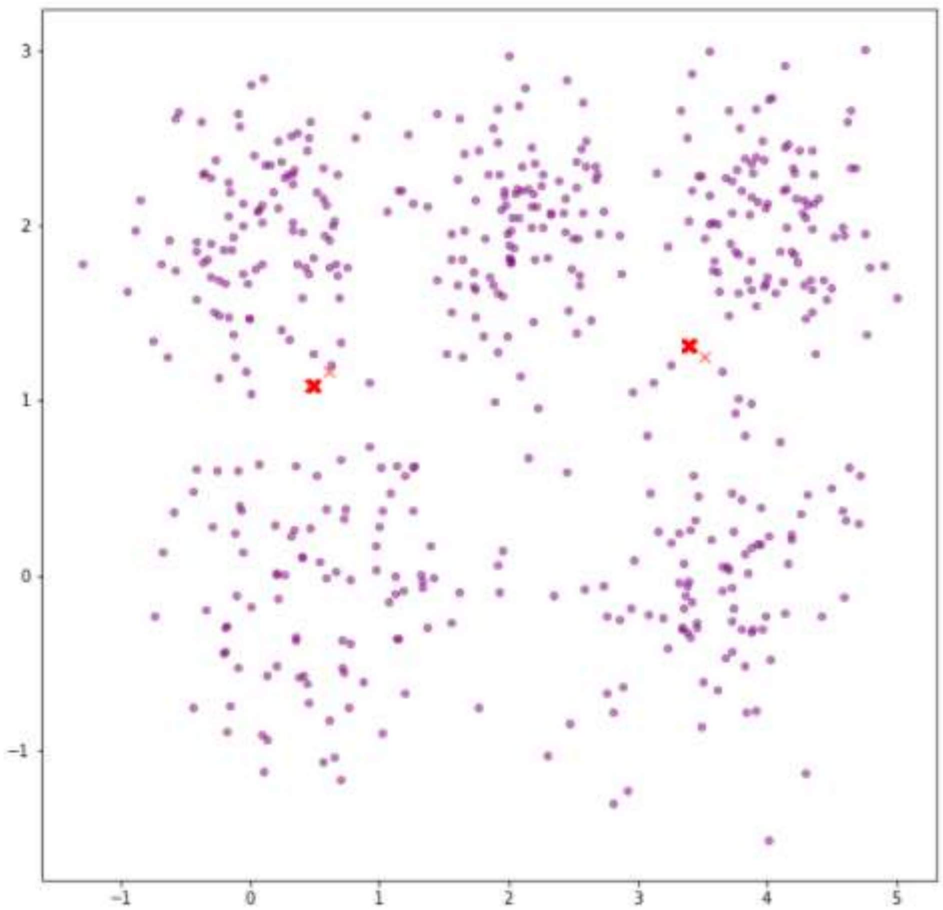
When I draw the data graph of the problem, the reason for this happens becomes clearer. In this data, there are three large clusters that are completely separate from each other, and if we continue clustering hierarchically from top to bottom or if we re-cluster on each cluster, each of these can be divided into three clusters. be divided again. That is, the data of the problem has an optimal clustering with 3 clusters and each of these clusters has 3 sub-clusters, all of which can be recognized on the shape of the data of the problem. Now, if we go back to the root sum of squares diagram with this explanation, the first point where the drop in value is severe is on 3, which is equal to the state where the algorithm detects the 3 main clusters. And if we look at the second point of the sudden decrease in the slope, this number is 9, which is equal to the state where the algorithm identified each of the sub-clusters we mentioned and now takes each one as a cluster.



And for example, a clustering with size 3 and 9 on this data:

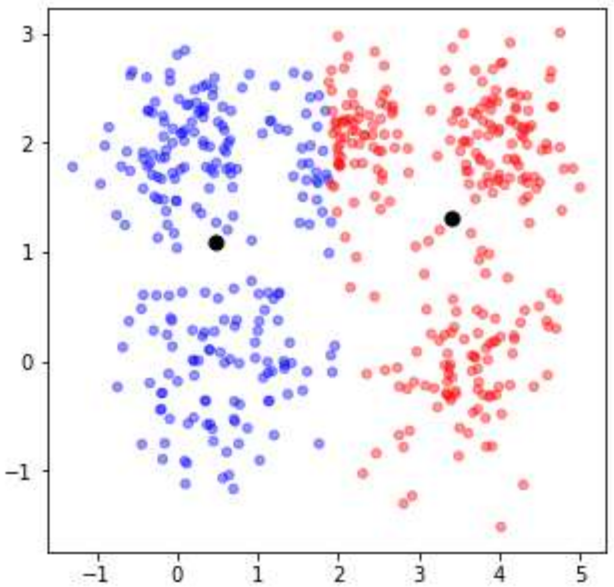


e) For this part, I repeat the same steps as part a for the data of the third_clustering_dataset.csv file. In this section, I make the number of clusters 2 and the number of clustering times 500 times with the random initial center selection method.

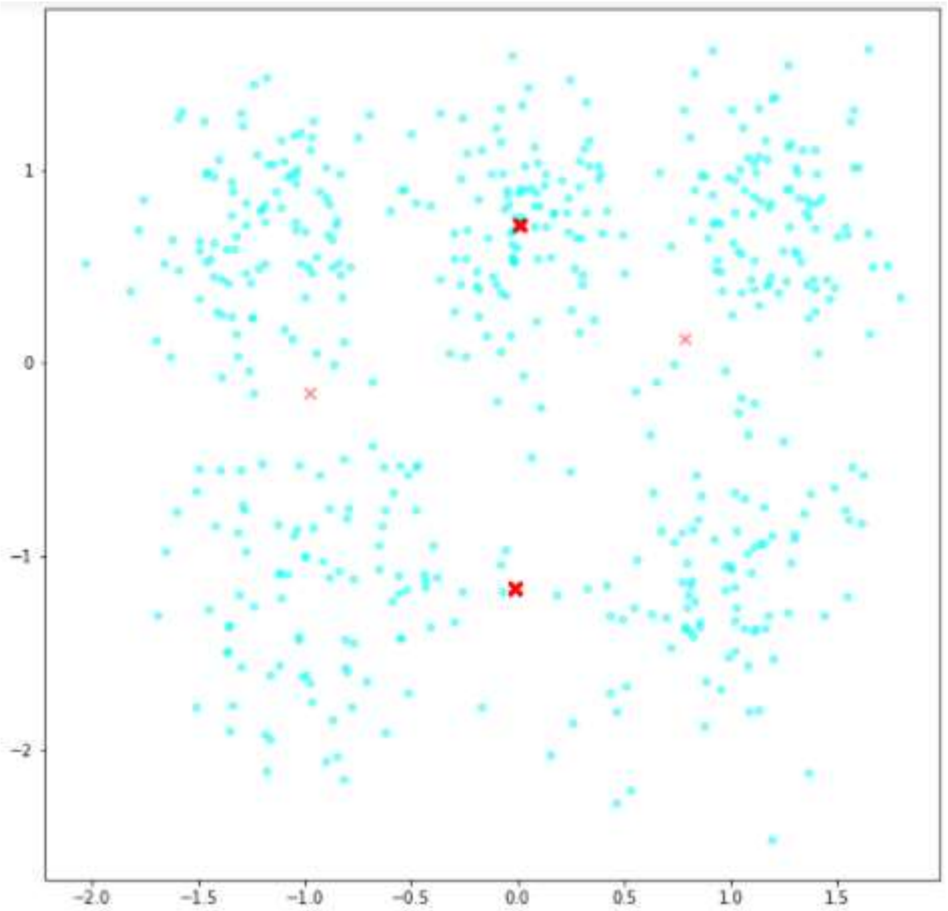


The 500 centers obtained for each cluster each time are so close that we can assume a point on this graph with this scale (with the exception of one case). These two points on the vertical axis are not very different from each other, so it can be said that the data are clustered in such a way that they are separated by the variable value of the first column. On the horizontal axis, we have a cluster on the left and a cluster on the right. If we pay attention to the data, a 5-level mode can be seen on the graph.

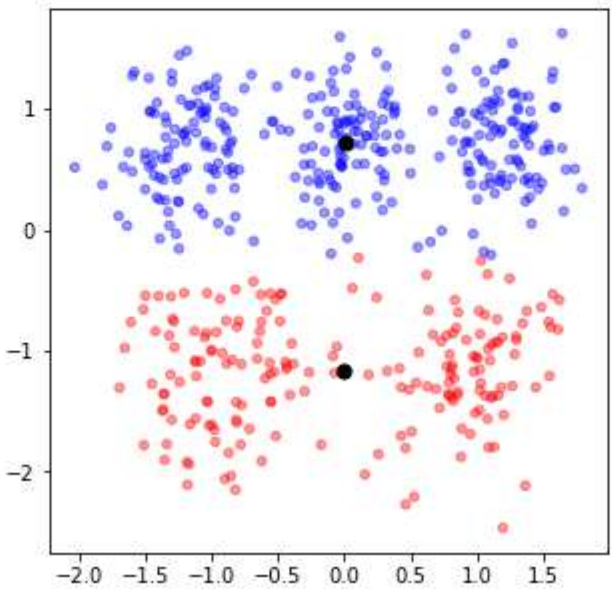
The image of the last clustering done:



f) I repeat part e this time with one more step and first I standardize the data with preprocessing. StandardScaler so that the mean is zero and the variance is one on each column.



There is no difference in the ratio of data to each other. The shape of the diagram is the same as the previous one and the same 5 clusters can be seen on it, and only the scale has changed, but if we compare the centers of the clusters to each other, in most of the implementations of this section, the position of the centers in relation to the rest of the data is as if compared to the previous state. It is on the imaginary line perpendicular to it. 2 clusters is not the best case for this problem, but with this choice in the previous case, it can be said that the data was divided into two clusters, left and right of the graph, and the two cases were divided into two clusters, upper and lower, after normalization. which seems to be a better result and is so obvious than the first case that it can be seen from the graph. That is, this time, unlike the previous section, the most determining factor for clustering is the direction of the variable, such as the second column of data. (of course, in one of the results, among the different implementations, the centers are similar to the section, and this may be due to the randomness of the model for choosing the initial center) It happened while they were naturally close to each other and it was better to fit in one cluster. But in the second case, after normalization, it seems more logical to imagine a hypothetical horizontal line separating two clusters on the data. Comparing the image of the last clustering model of this part with e confirms the same result.


1
0
−1
−2
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5