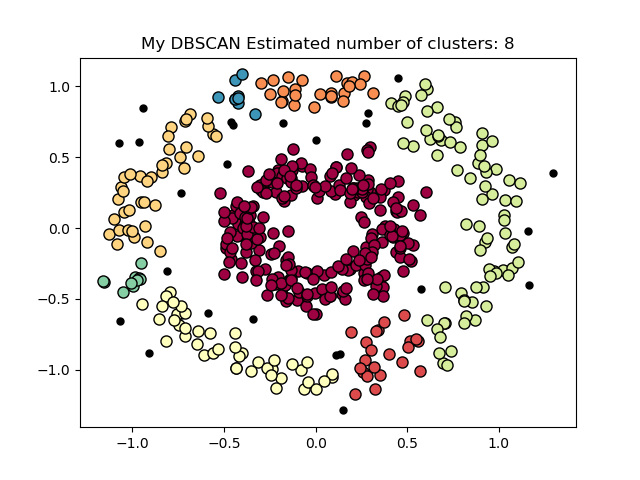
**DBSCAN**

For the following scatter plots:

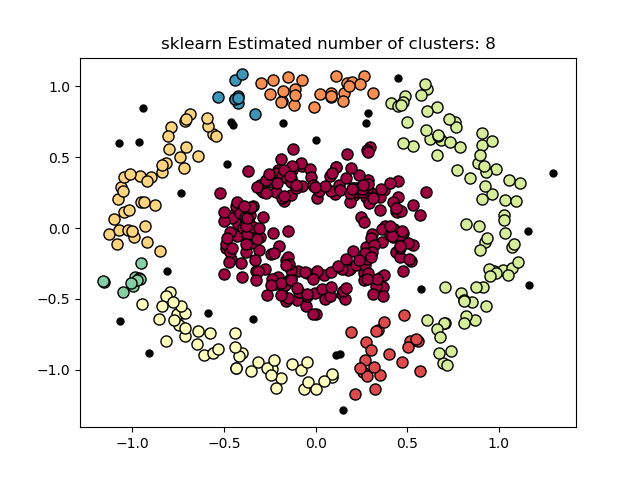
* Nodes of the same color have the same label
* The smaller black points are noise points

**Default Setting**

My Results

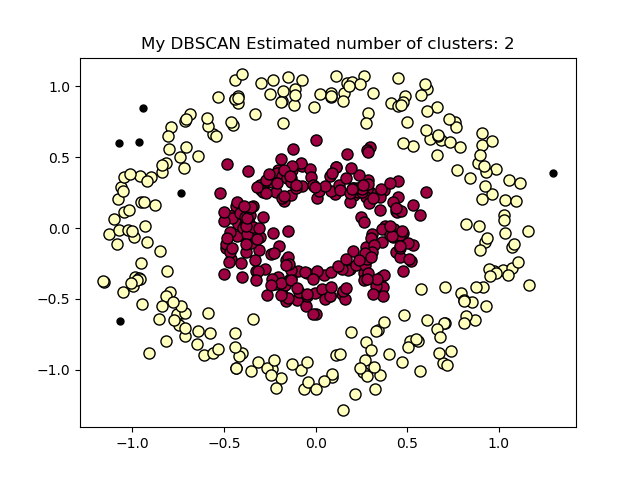


Sklearn Results

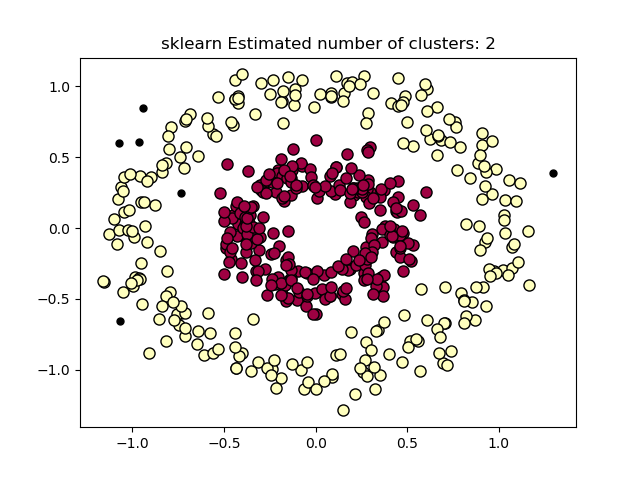


**eps increased to 0.15**

My results

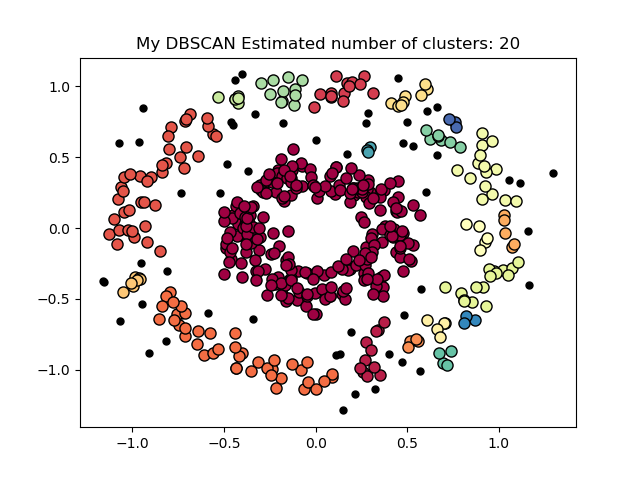


Sklearn Results

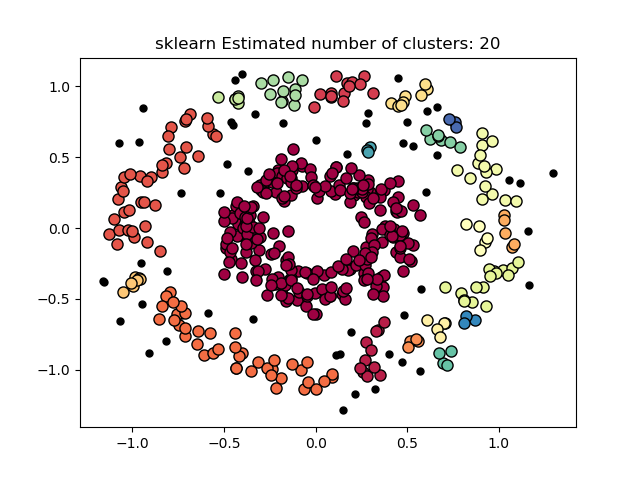


**esp decreased to 10**

My Results

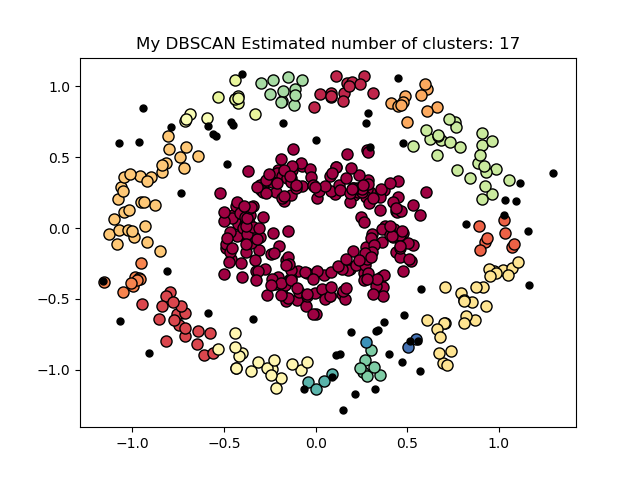


Sklearn Results

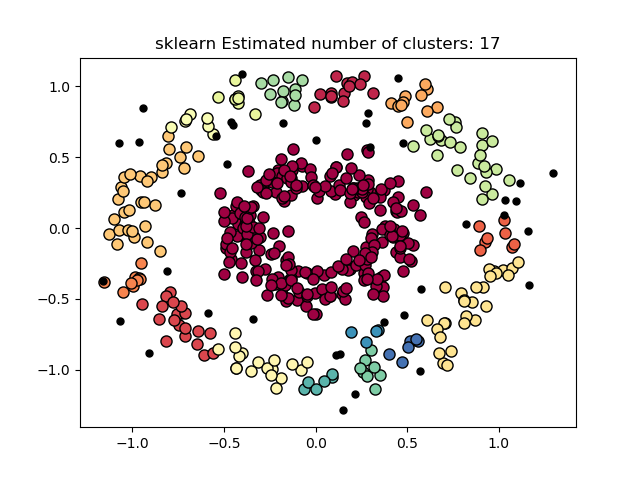


**min\_samples increased to 5**

My results

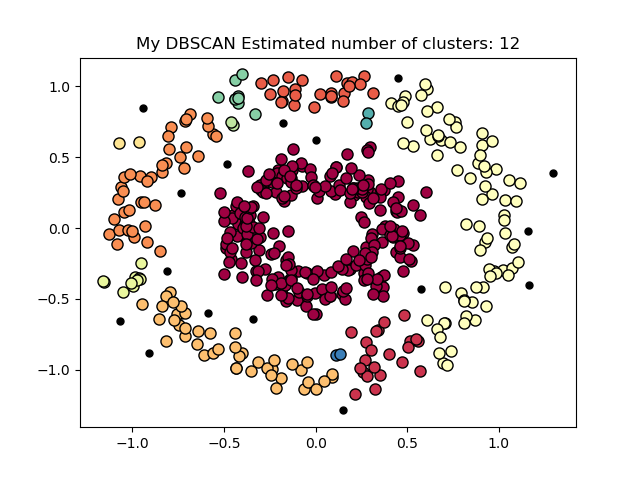


Sklearn Results

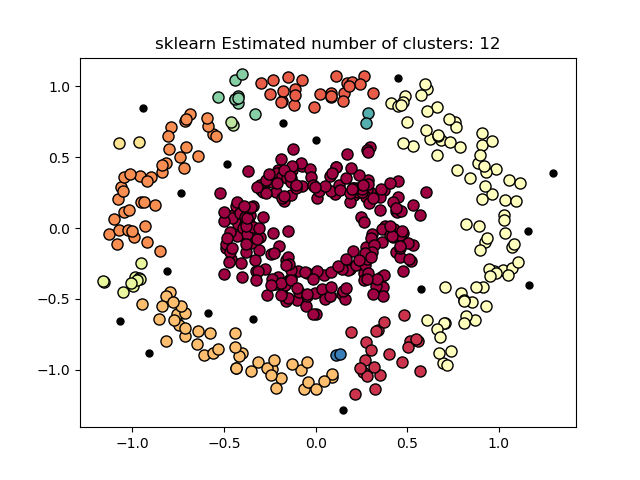


**min\_samples decreased to 2**

My Results



Sklearn Results



**Conclusion:**

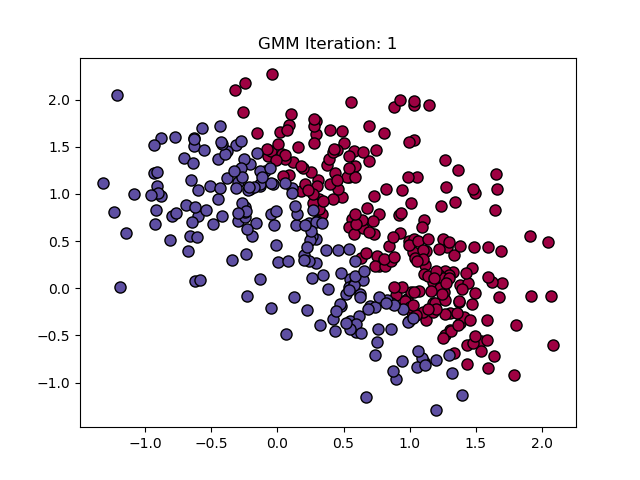
My results were almost identical to the results produced by the sklearn package. Throughout each test I was able to produce results that gave an identical number of labels and draw a cluster that looks indistinguishable to the sklearn results. However, for some of the tests, when directly comparing labels, the results differed slightly. This is likely due to a difference in the method that was used to calculate the Euclidian distance between two points.

When I increased the esp variable there was a decrease to the number of clusters and when I decreased esp the number of clusters would increase.

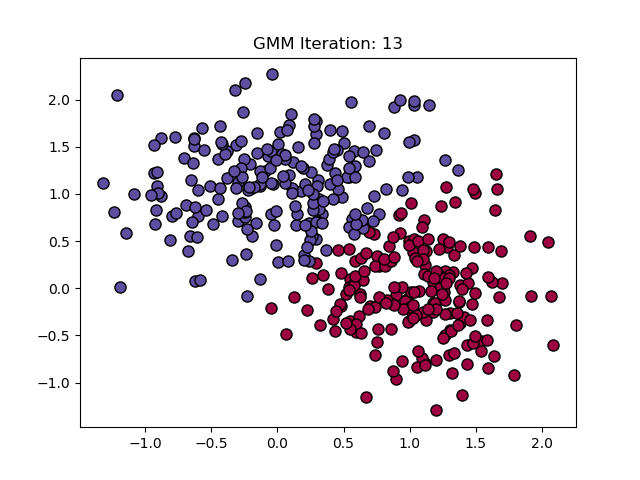
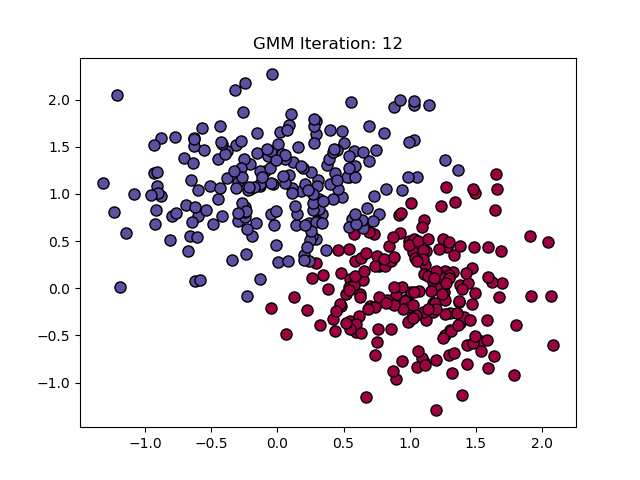
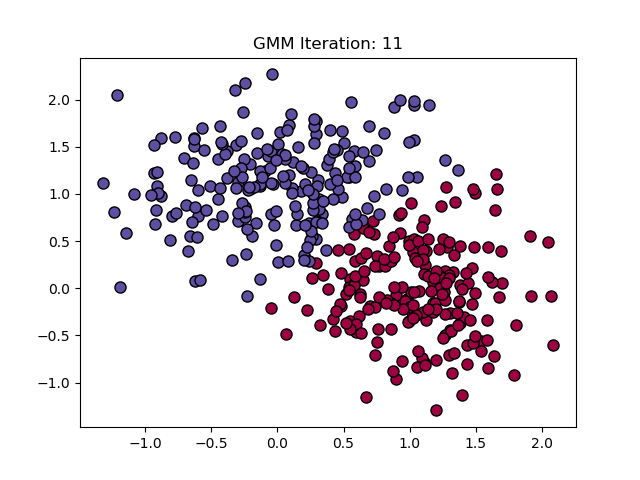
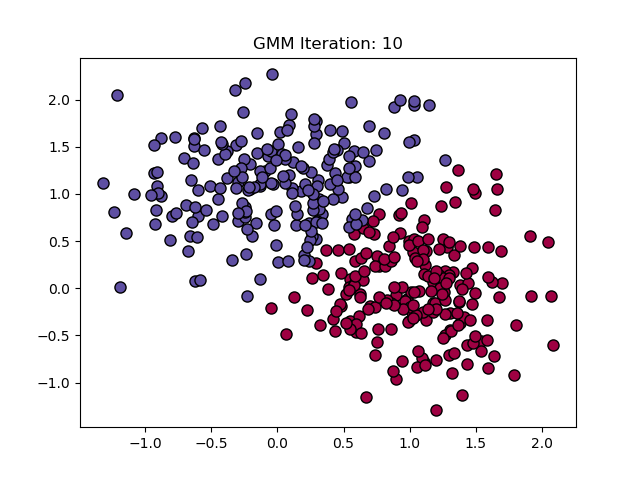
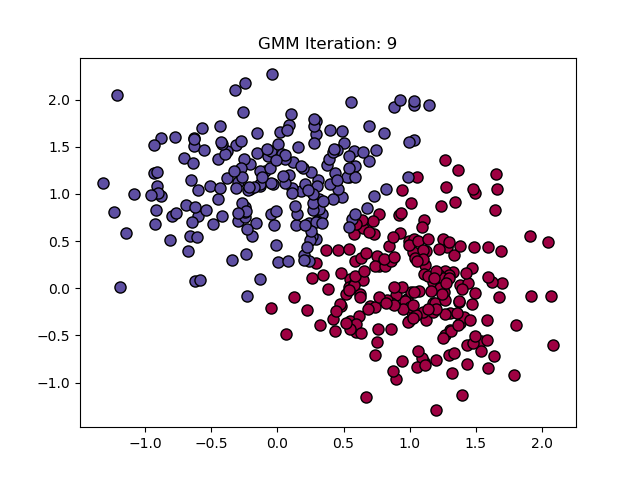
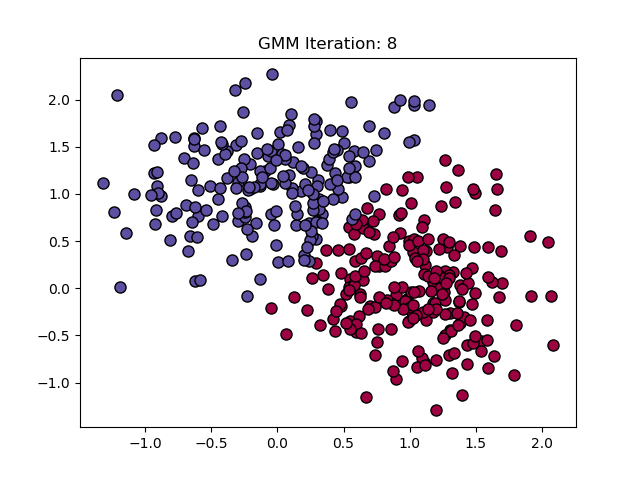
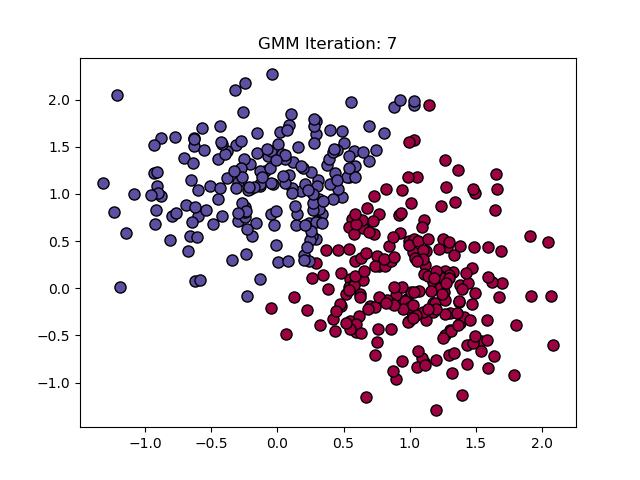
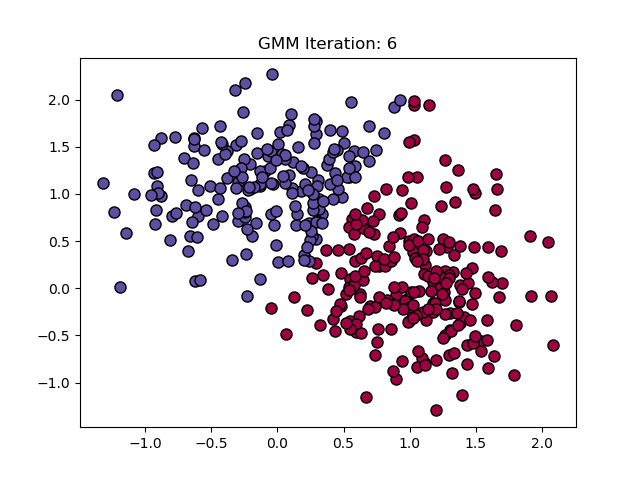
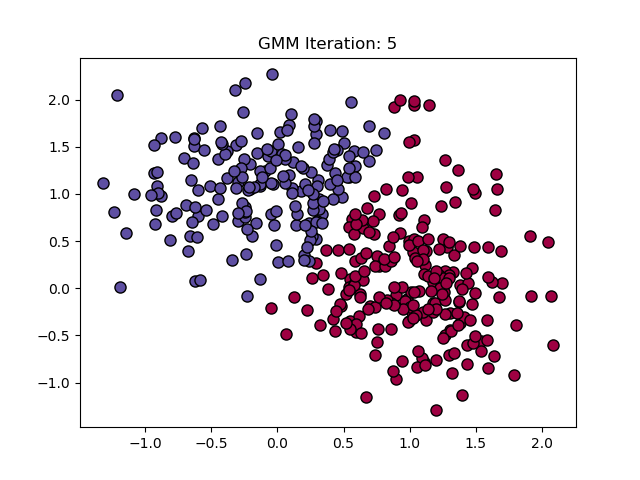
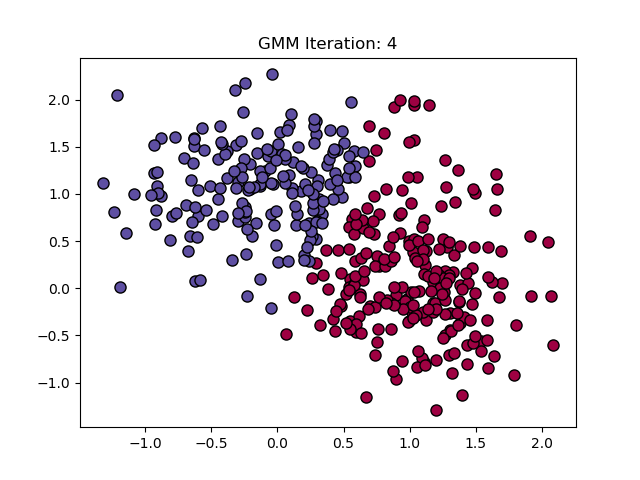
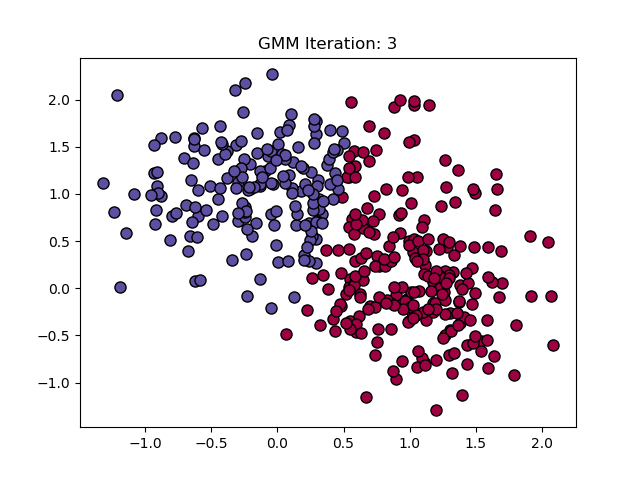
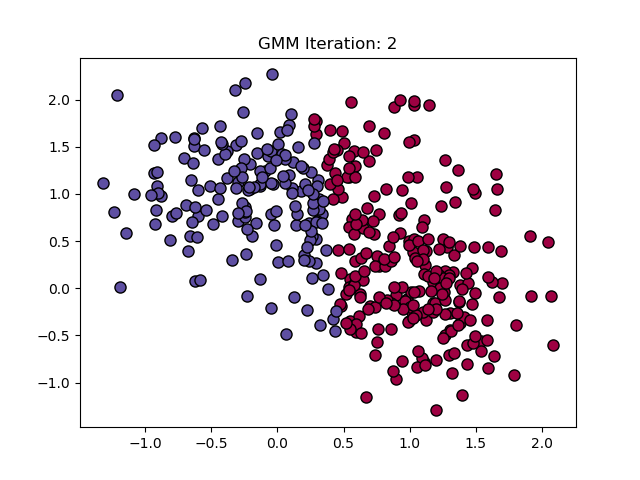
The number of clusters increased when I increased and decreased the min\_sample requirement. The increase in clusters was less significant when I increased min\_samples compared to when I decreased it.

Based on these results, we can conclude that the number of clusters is greater when the esp is high compared to when its low. The number of clusters with regard to min\_samples is more complicated and likely depends more on the sample data.

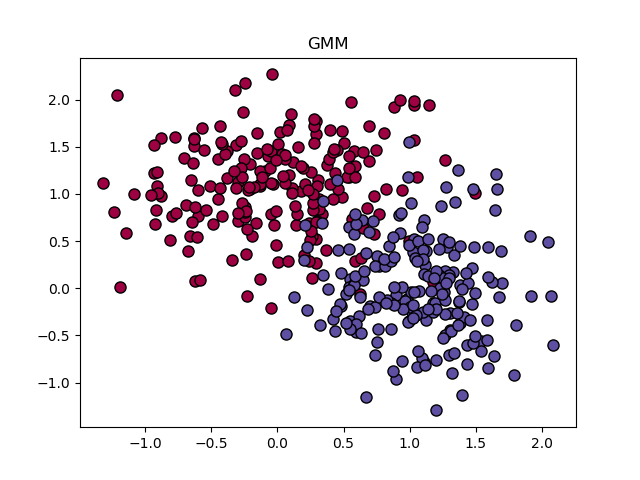
**GMM**



My Results



True Labels



**Conclusion**

Initial Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Mean 1 | Mean 2 | Std. 1 | Std. 2 | Lambda |
| x | 3 | 0 | 1 | 1 | 0.5 |
| y | 0 | -2 | 1 | 1 | 0.5 |

Final Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Mean 1 | Mean 2 | Std. 1 | Std. 2 | Lambda |
| x | 1.062527 | -0.001631 | 0.402160307746192 | 0.5442510005272795 | 0.4875 |
| y | -0.038133 | 1.108867 | 0.4647083345643792 | 0.4514380115769025 | 0.5125 |

My algorithm ran for a total of 13 iterations before finally terminating. The final result gives us a scatter plot that is noticeably similar to the plot of the true labels. However, my algorithm was not able to find all the true labels. In the true labels plot, there are a lot of points in both clusters that are overlapping. There are many points from one cluster in the middle of the other one. My algorithm was unable to label those points properly, instead forming a clear divide between the two clusters.