Kolby Sarson
104232327

Assignment 2

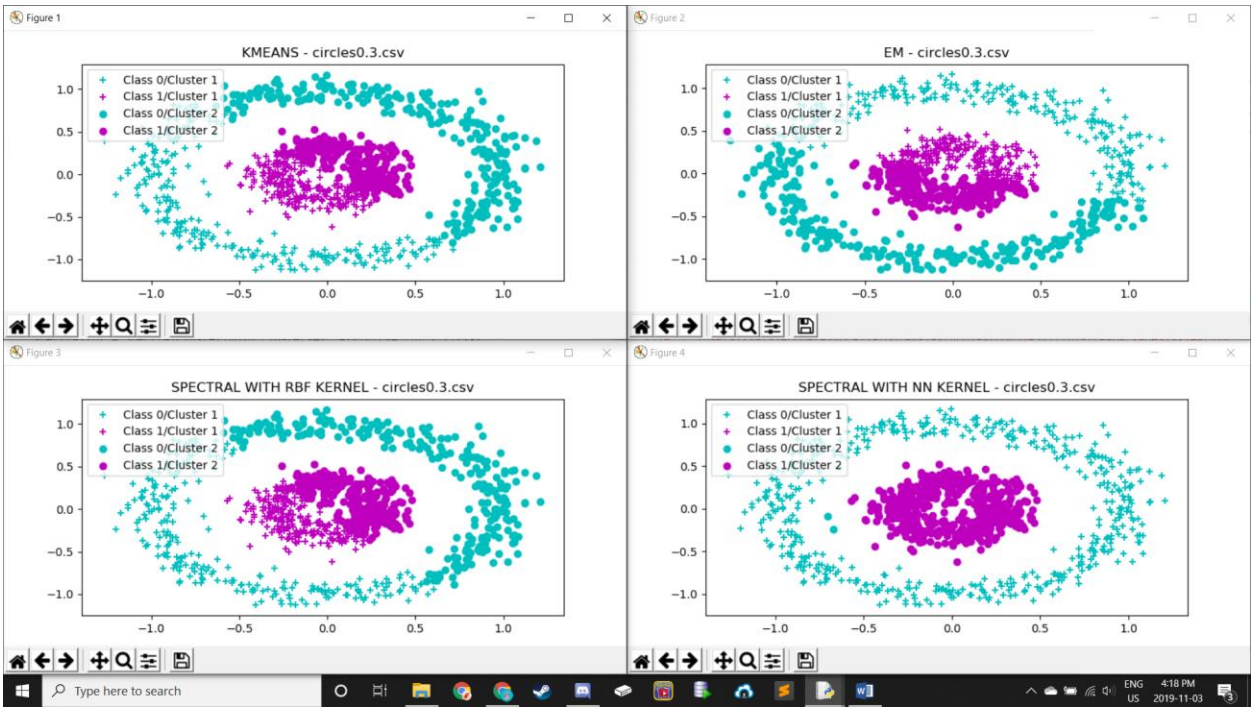
Running SciKit-Learn

To run the clustering algorithms on SciKit-Learn I first imported 'pandas' to deal with the .csv files, 'pyplot' to plot the data, and the clustering algorithms 'KMeans', 'Spectral Clustering' and 'mixture'. Next, I created an array for the four classifiers in order to loop through each classifier for each dataset. I then implemented a loop to account iterated through all the datasets and used an if statement to account for 'spiral1.csv' having different column labels. Inside the loop I initialized the clustering algorithms and use them to predict the new labels and inside the if statements I group the old .csv labels and new predicted labels into a new data frame. Finally, I use this new data frame to plot the graphs.

2.

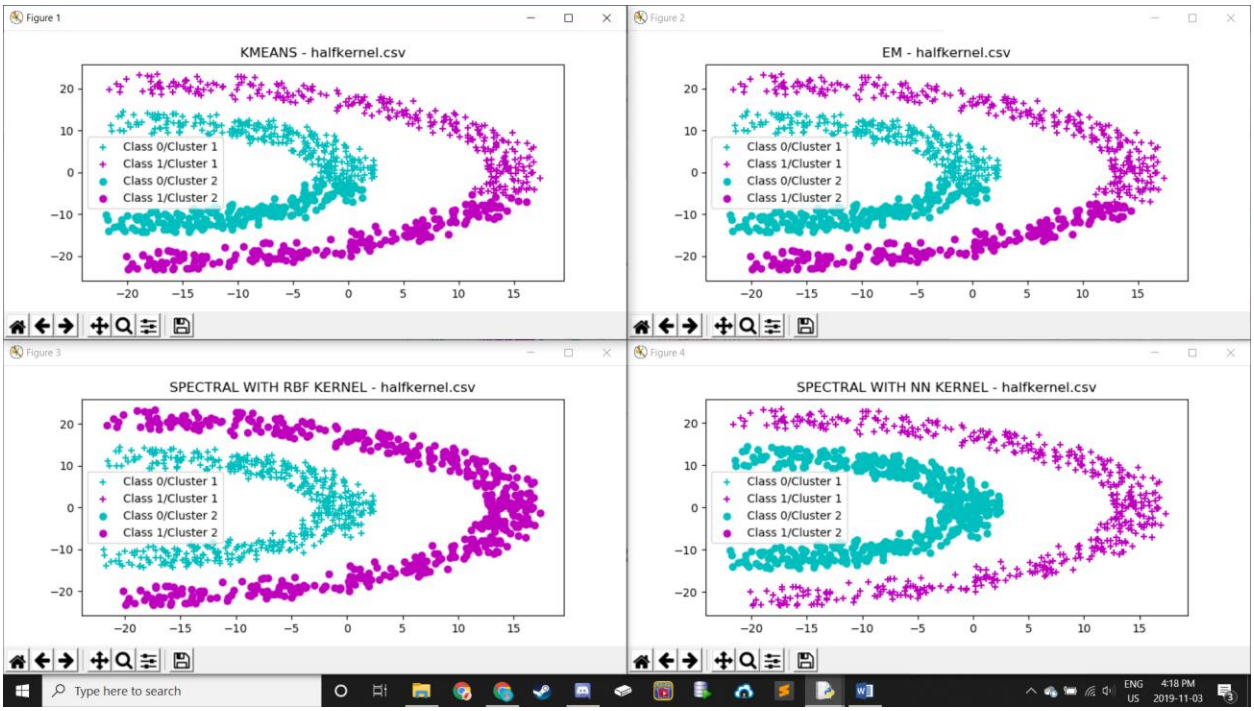
a. Choose a distance function and explain why you chose it.
   Scikit-Learn currently only supports Euclidean distance so for that reason I used this distance function in my K-Means clustering.
c. Choose a kernel and number of neighbors and explain why you chose them. Suggestion: use RBF
   Scikit-Learn supports both RBF and Nearest-Neighbour kernels. When using RBF, the number of neighbours plays no part in the clusters whereas in Nearest-Neighbours it does. I chose to implement both kernels in my code so that I can look at both side by side and see the difference in performance for each dataset. As would be expected, each of RBF and Nearest-Neighbours had datasets where one outperformed the other, so certain datasets could benefit more from one or the other, but neither is better all the time. As for the number of neighbours chosen, I opted to use k=10 as this is the default. We will see which kernel performed better for each dataset as we progress through the report.

Kolby Sarson
104232327



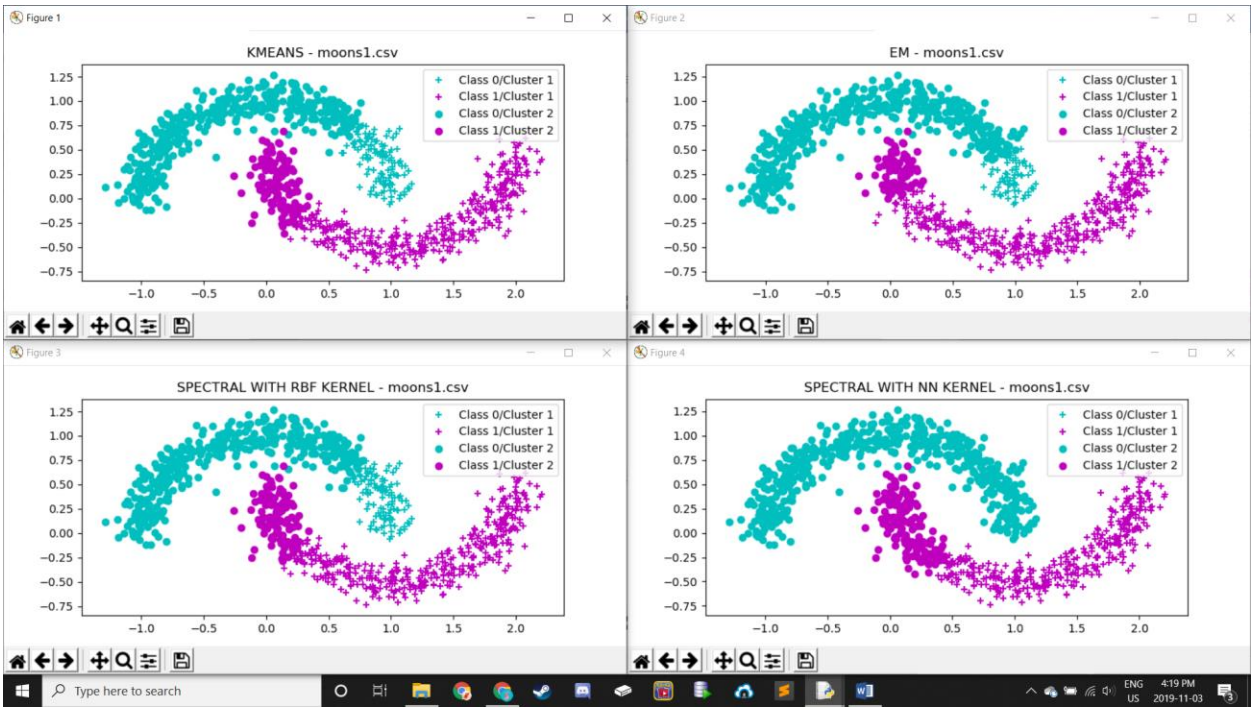
In this image we see that K-Means, EM and Spectral Clustering with RBF all perform relatively poorly. They all decide to divide the circle in half to produce the two clusters, indicating approximately 50% accuracy. Once we implement Spectral Clustering with NN, we see much greater results, nearly 100% in fact, making Spectral Clustering with NN the best clustering algorithm for this dataset.

Kolby Sarson
104232327



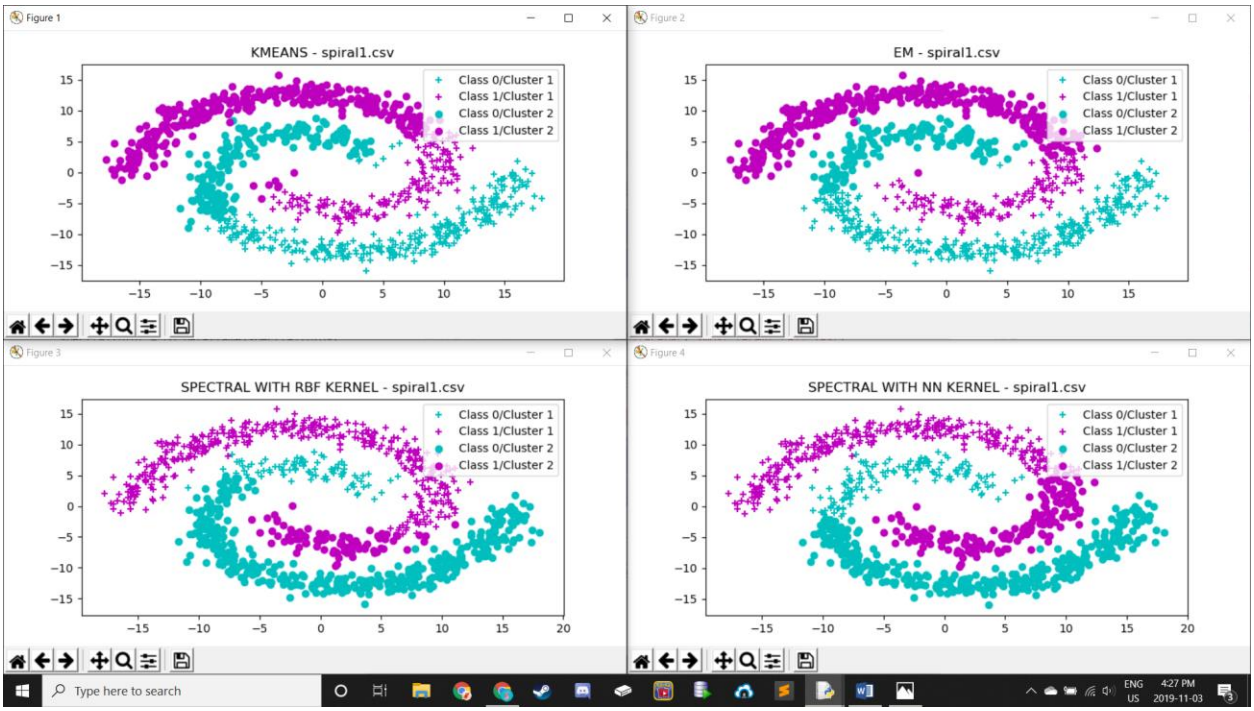
In this image we see that K-Means and EM both perform relatively poorly again, following the same ideology of dividing the data in half, producing approximately 50% accuracy. However, both forms of Spectral Clustering, using RBF or NN, seem to be effective with nearly 100%. Therefore, either of these clustering algorithms would be effective for this dataset.

Kolby Sarson
104232327



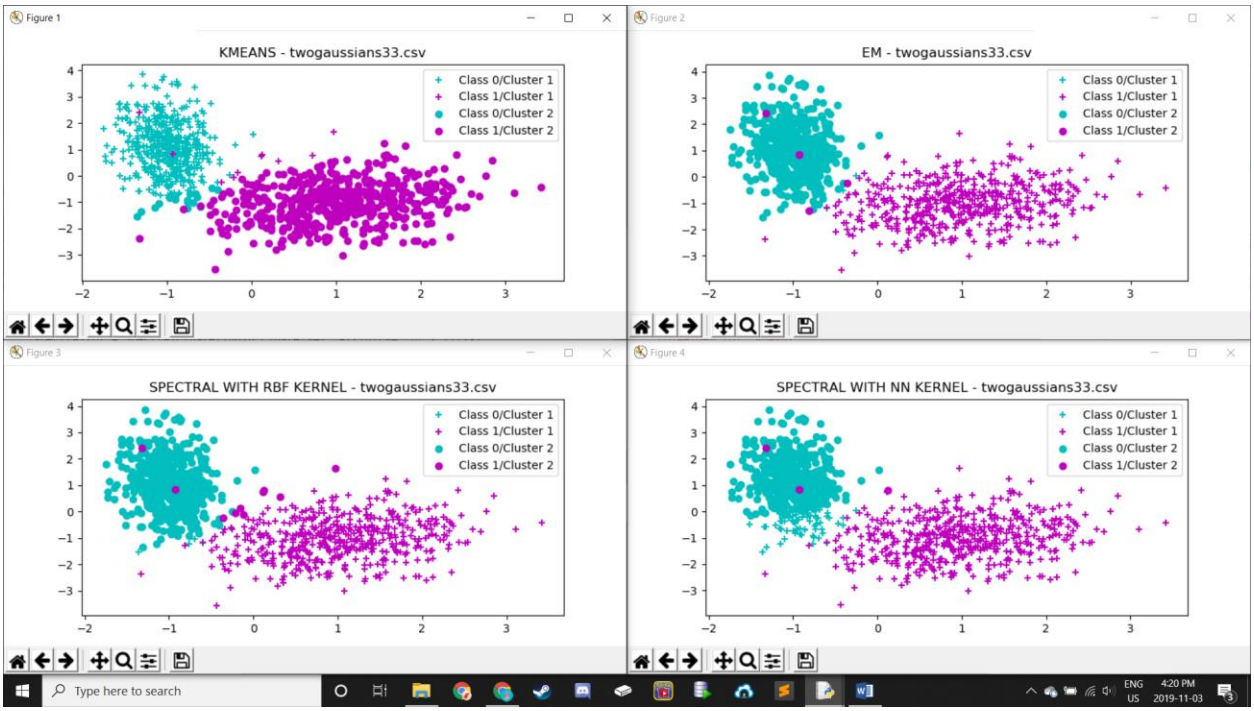
Once again, we see that K-Means and EM perform poorly. However, we now see Spectral Clustering with RBF also perform poorly for this dataset, presumably due to its use of a fixed point. A fixed point would not be able to be determined that would help cluster the data efficiently. Now looking at Spectral Clustering with NN, it performs almost the same as using RBF for the first cluster but has nearly 100% accuracy for the second cluster with respect to Class 0, therefore outperforming Spectral Clustering with RBF.

Kolby Sarson
104232327



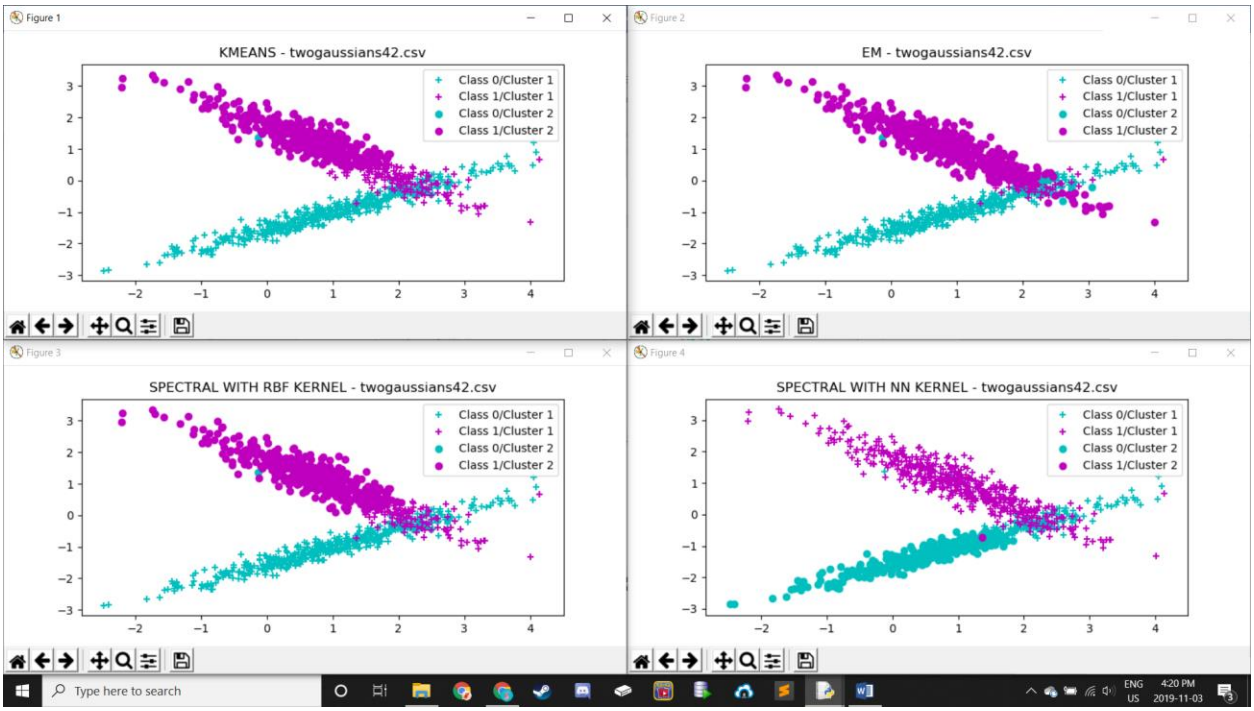
Yet again, K-Means and EM underperform with this dataset, dividing the data in half again. Interestingly, with this dataset Spectral Clustering with NN seems to also simply divide the data in half, assumed to be caused by the fact that the data is almost ‘intertwining’ with each other. The best performer for this dataset seems to be Spectral Clustering with RBF with approximately 75% accuracy.

Kolby Sarson
104232327



This dataset is one which appears to be almost perfectly linearly separable, which results in all our clustering algorithms having high performance. EM appears to be the optimal solution here with only a few outliers or overlapping data failing to cluster properly. This is closely followed by Spectral Clustering with RBF. Our remaining two clustering algorithms, K-Means and Spectral Clustering with NN both perform highly but are noticeably less effective than the first two algorithms mentioned.

Kolby Sarson
104232327



With this dataset, EM appears to be the best option with only a few data points being clustered incorrectly. The remaining three datasets all seem to end up dividing the data linearly, which is not a bad option for this dataset, but is also not the best option. As soon as the intersecting data is reached, these algorithms assign all of the data from the intersection onward to one single cluster, which is effective for one cluster but for the other cluster is only effective until this intersecting data is reached.