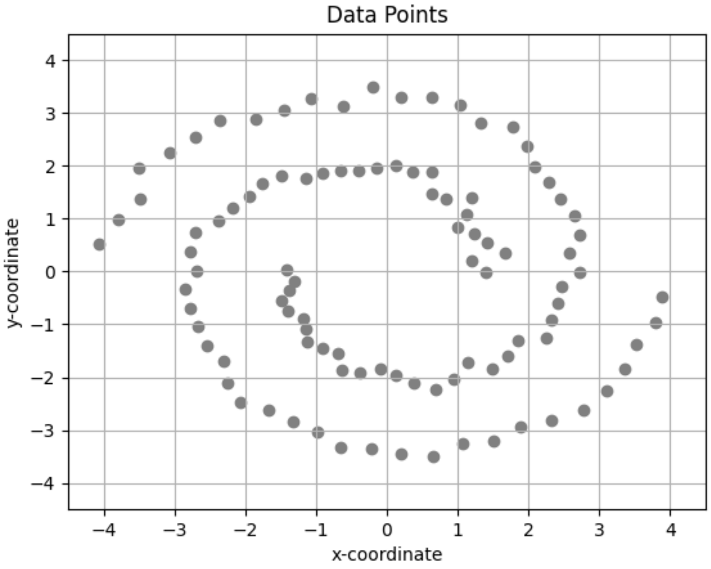
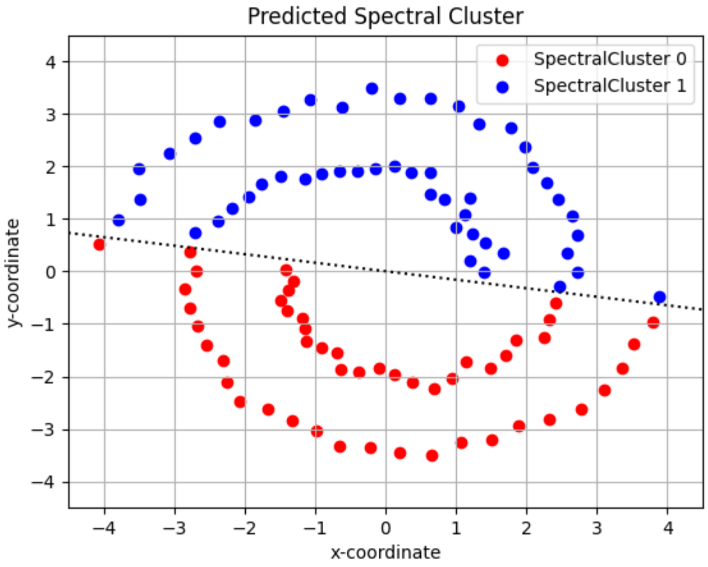
Cameron Cooke

This project comes from an assignment in my Intro to Machine Learning course taken in Fall 2021. The purpose of the assignment was to study various clustering algorithms, this project in particular studying the Support Vector Machine (SVM) algorithm. We were given a dataset with 100 values, and with Python and the Sci-Kit Learn library, we used the SVM algorithm to classify the values into appropriate clusters, with the goal being to use these clusters to classify new values. This writeup walks through that process.

Raw dataset: Ideally, the two spirals should each be classified as a cluster.

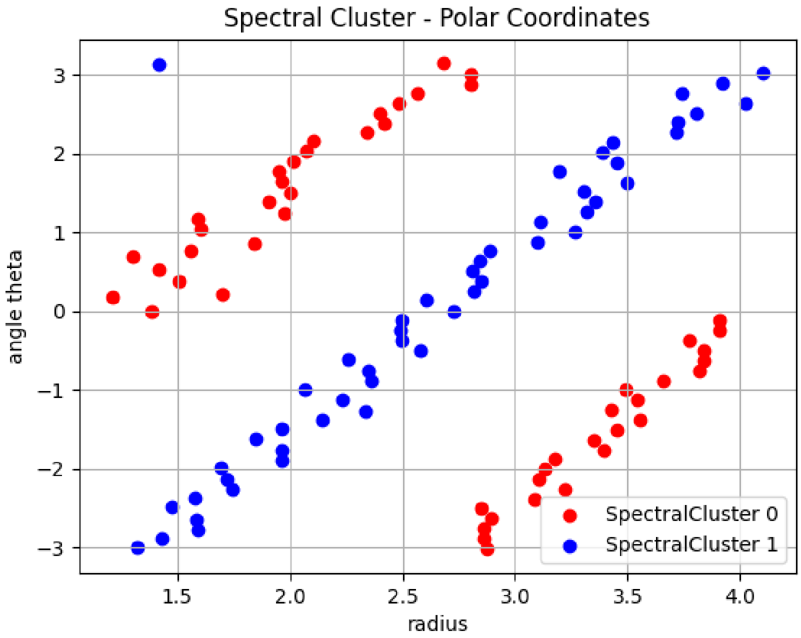


Step 1: Attempt to run Python's sklearn.svm.SVC function on the raw dataset. The classification can be seen in the following graph. As you can see, this results in a misclassification rate of 50%.

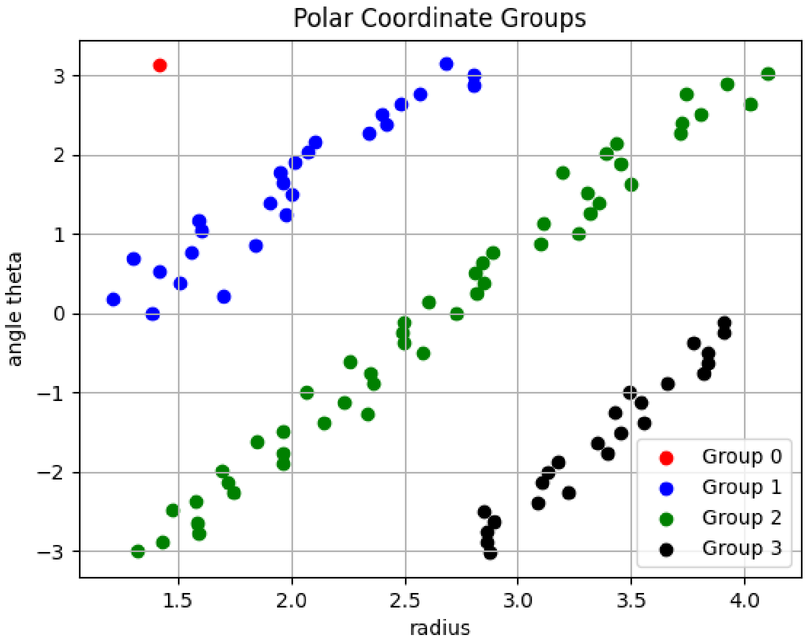


Cameron Cooke

Step 2: The values cannot be accurately separated by a hyperplane (dotted line) from the SVM algorithm, so the data can be converted into polar coordinates to help. The following graph shows this conversion to polar coordinates.

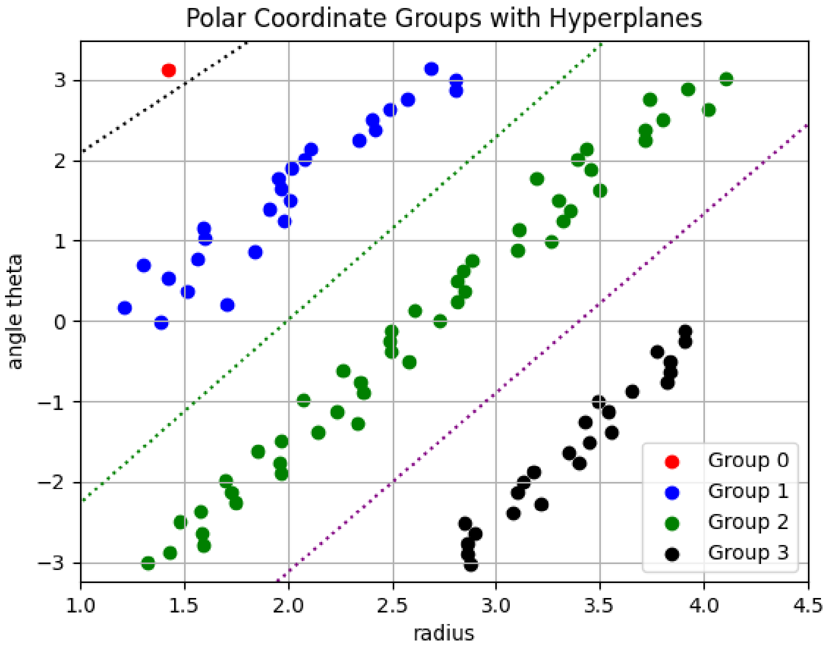


Step 3: Now the data can clearly be separated into four groups. The following graph shows how these four groups were separated.

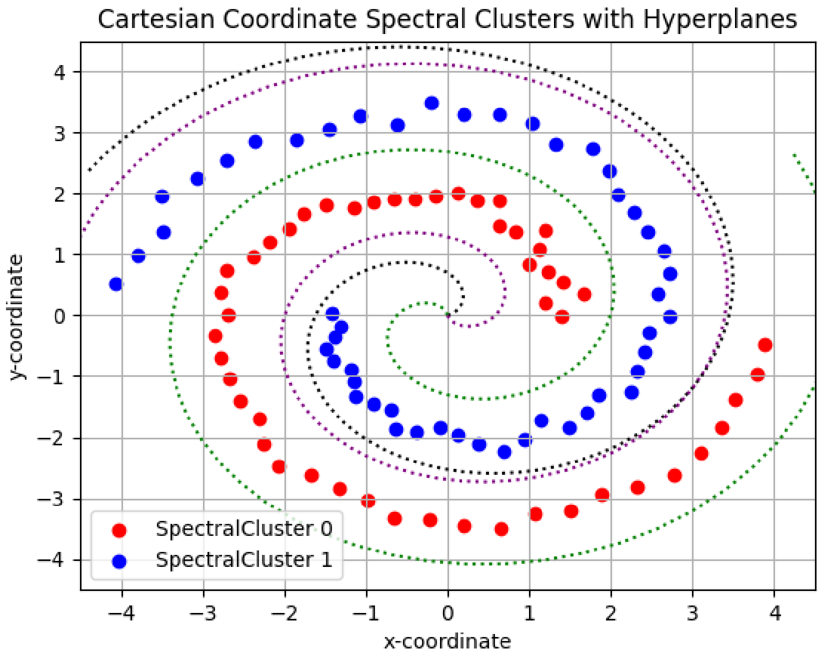


Cameron Cooke

Step 4: Run the SVM algorithm 3 times, once on each neighboring group, to determine the separating hyperplane between each of them. The following graph shows these hyperplanes (dotted lines).

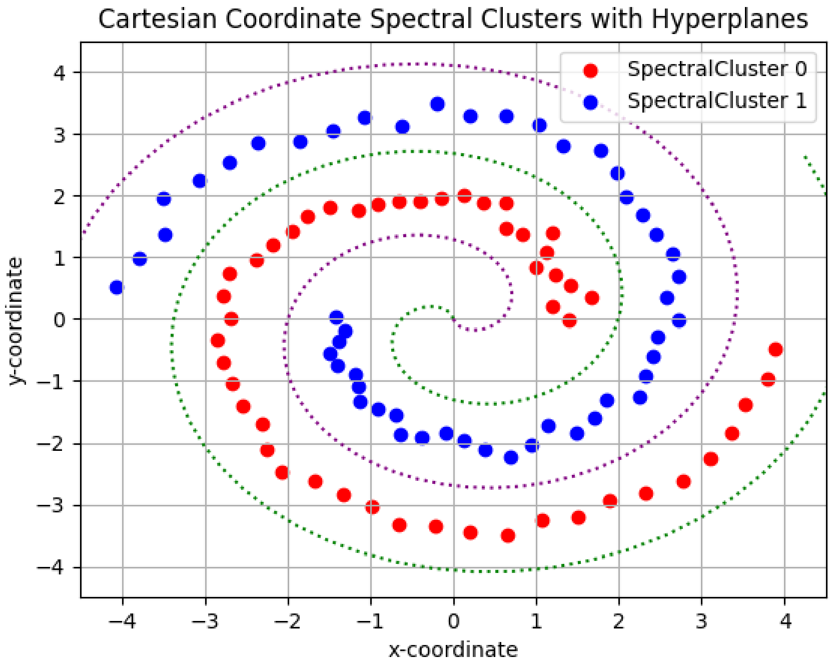


Step 5: Convert the observations and hyperplanes from polar coordinates back to cartesian coordinates. The following graph shows the hyperplanes separating the data in the original cartesian coordinates.



Cameron Cooke

The black curve is not needed, as it effectively does the same thing as the purple curve, and it was only calculated to cover a single datapoint, so it can be omitted. The result is a graph of the dataset properly classified into two clusters. We now have a means of accurately classifying new values.



The code for this project can be found in SVM.py.