Intelligent Systems 2 Final Report

AI Cluster Algorithm Summary

For my assignment I chose to demonstrate a few clustering algorithms on a chosen datasets. I used three different clustering algorithms; these 3 are Affinity Progression, K-Mean and Spectral Clustering. I am aiming to show the accuracy and usability of different clustering algorithms in regards to the Iris Datasets, the job of the algorithms is to predict which data points are parts of which species.

Affinity Propagation is a clustering method that differs from most other clustering methods as it requires no input of specificity on how many clusters, this means that the algorithm will determine itself how many clusters there are. How the Affinity Propagation works is that each individual data point sends messages to each other individual data point, as well as replying to each message, telling them how far away they are from each other. From this messaging and coupled with a preference the algorithm then puts each data point in its respective cluster; this preference controls the amount of clusters that the algorithm creates, by having a lower number (ie. -50) you get less clusters and the closer the preference is to 0 the more clusters it will create with 0 being when each individual data point is its own cluster, the preference effectively is the maximum negative square Euclidean distance that a point can be away from the exemplars, exemplars being the centre of the cluster.

K-Mean is a widely known clustering algorithm that will put the data points in a number of clusters equal to K. how this algorithm works by when you select a value for K it then selects K random data points as the initial clusters and it then measures the relative distance each other points are to each cluster and assigns them to the closest one. After this it then takes the median point between each point in each cluster and repeats the previous steps but with the medians as the new cluster points, it then repeats this until the clusters don’t change anymore. What then happens is that it calculates and records the sum of the variations of each cluster, the variation is the greatest distance between two points, to find the total variation, it then repeats the initial steps with new random points and records the total variation and compares it to the previous total variations, and once all combinations of initial clusters have been exhausted it returns the shortest variation, this provides the optimal clusters for the dataset. Spectral Clustering is similar to K-Means however it does not assume the shape of the cluster where most algorithms will assume that the clusters will take on a spherical shape, this means that clusters can take almost whatever shape such as o shapes with nothing in the middle or another cluster in the middle.

Affinity propagation is a useful algorithm to use when you do not have or do not know how many cluster you need as it determines this for you, this means that it actually isn’t greatly suitable for the data set I chose as I know that I will need 3 clusters as there is only 3 separate species of iris in the dataset, however it is informative to see how this algorithm would cluster and group each individual data point, as from my results I can see that it does wrongly classify points that differs from the K-Means and Spectral Cluster. The K-Means is definitely the most widely used clustering algorithm and it is so because it is very easy to implement as well as being generally very reliable in its ability to form clusters however in my case where I am trying to predict the possibility of the clustering being correct in defining the data points to the right species cluster it isn’t the best.

Figure 1 above and figure 2 below show the original Iris Dataset in graph form with all of the each individual plot points. From the graphs in figure 1 and 2 you can see the 3 original clusters which are the correct; each colour represents a different species of Iris from the datasets, you see more clearly in figure 2 that 2 of the species overlap significantly. This overlapping points means that the algorithms that I chose will incorrectly class certain data points into the wrong clusters as seen in figure 2 below on the right hand side, this is because the algorithms work on defining points which are close together. The K-Means algorithm on the right hand side of figure 2 shows the 3 predicted species clusters where they think that these data points belong to, you can see clearly defined clusters which are unlike the real clusters which has overlapping data points, therefore show the downfall of using the K-Means algorithm.

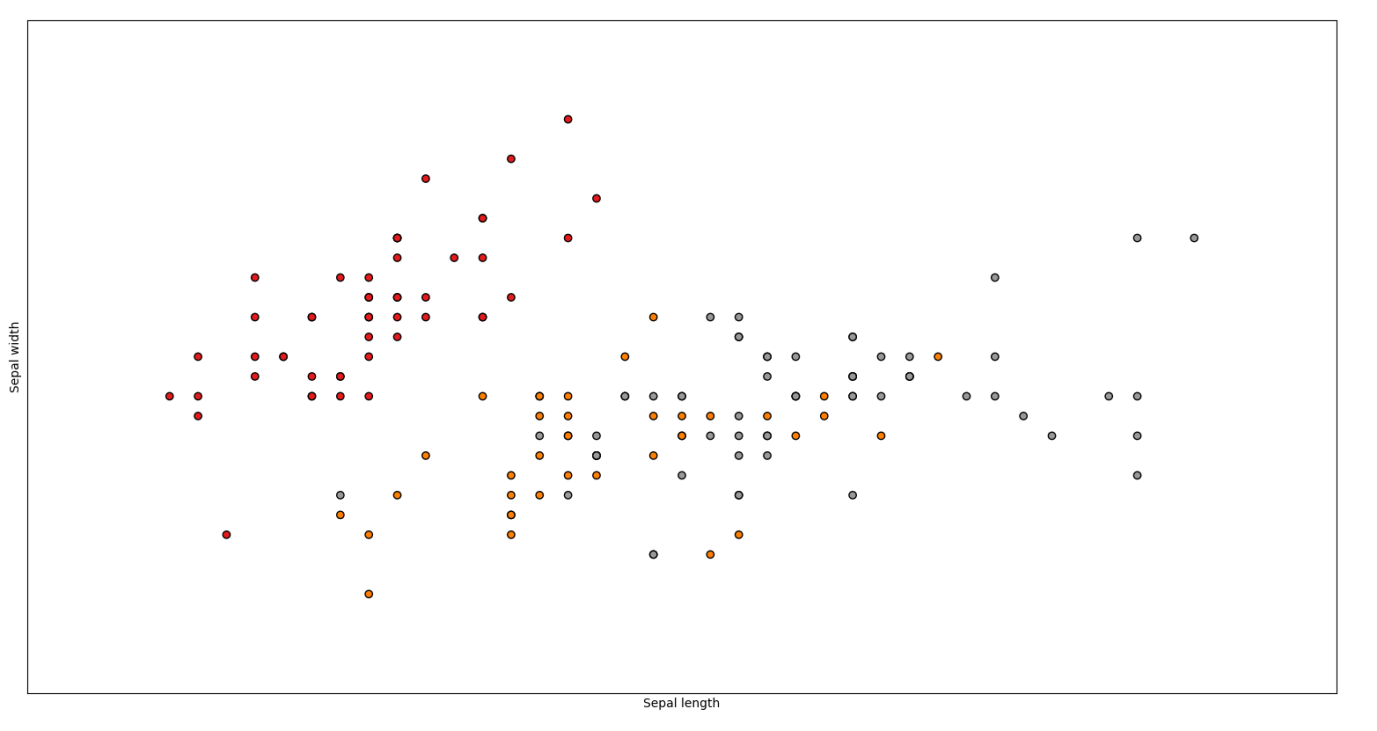


Figure : Iris Dataset

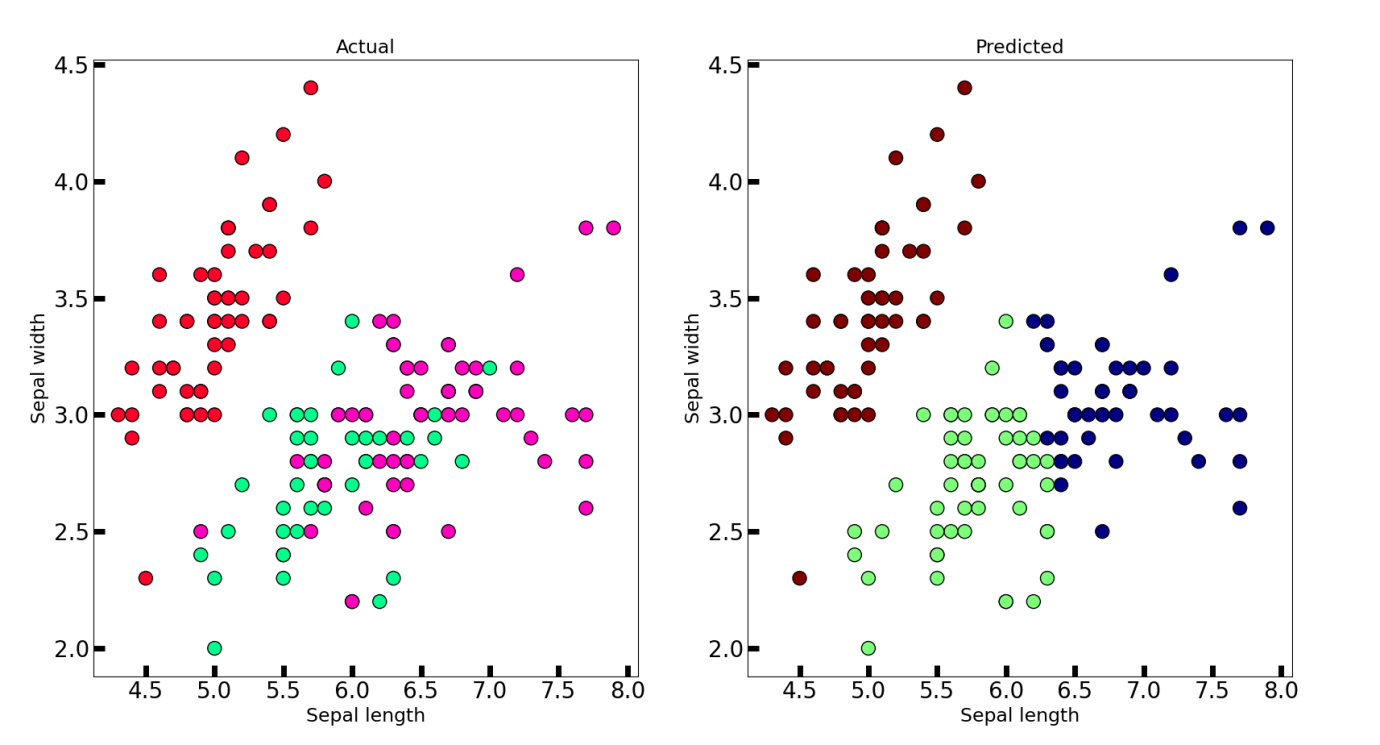


Figure Iris Dataset (left) and K-means Clustering (right)

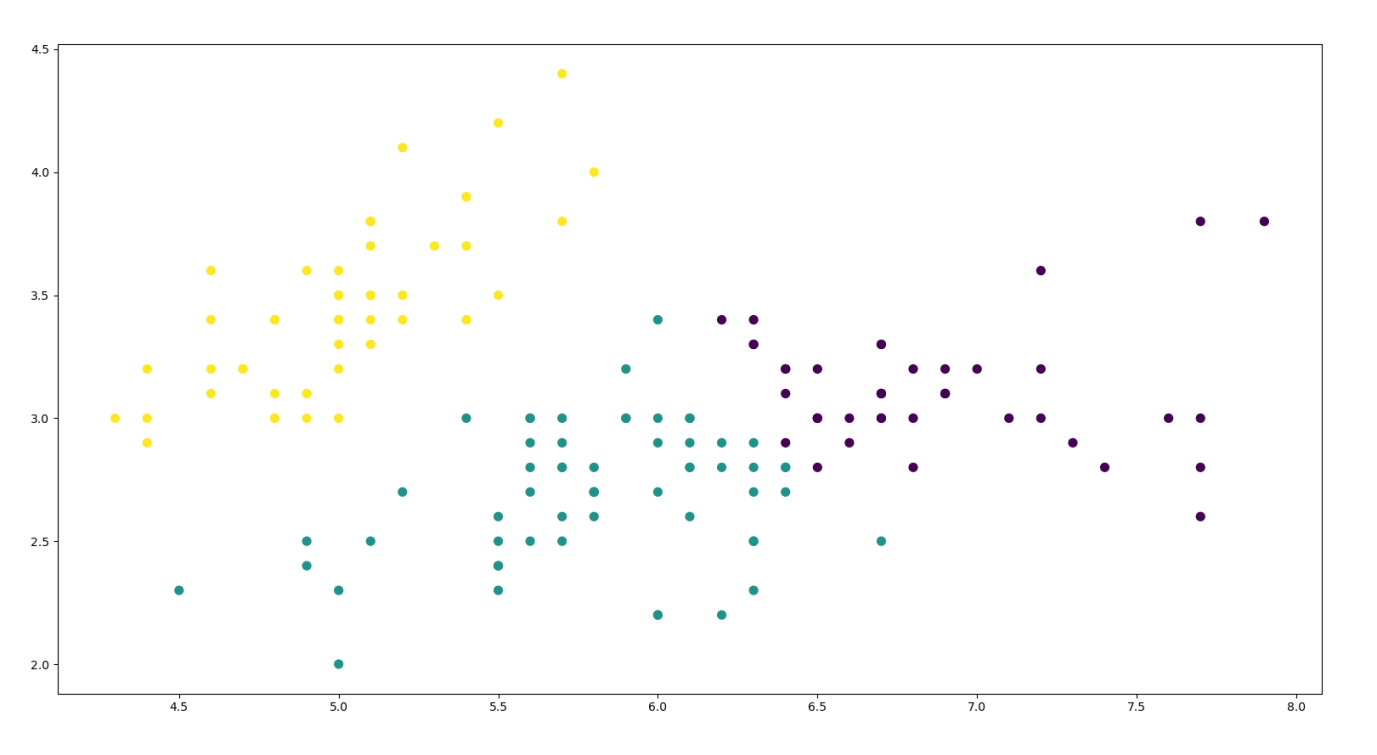


Figure : Spectral Clustering algorithm

In figure 3 it shows the predicted clustering of the datasets using the Spectral Clustering algorithm and comparing this graph to the K-Means you can see that there are a few minor differences, these differences show a few points have changed cluster, but overall still very similar with clearly defined clusters and no overlapping points. This algorithm didn’t work to its full potential because the dataset is fairly regular meaning that the algorithm doesn’t detect non uniformed shaped clusters. Figure 4 below shows the graph that was created when I implemented the Affinity Propagation clustering algorithms, this shows a greater difference in how the clusters should look compared to the original, K-Means, and Spectral Clustering as the right hand most cluster has significantly more data points attributed to it than the other clustering algorithms show. The graph itself is also different to the previous graphs as the Affinity Propagation algorithm has an exemplar in each of its cluster, which helps show how the clusters are connected, whereas the previous graphs have only different colours denoting the different clusters.

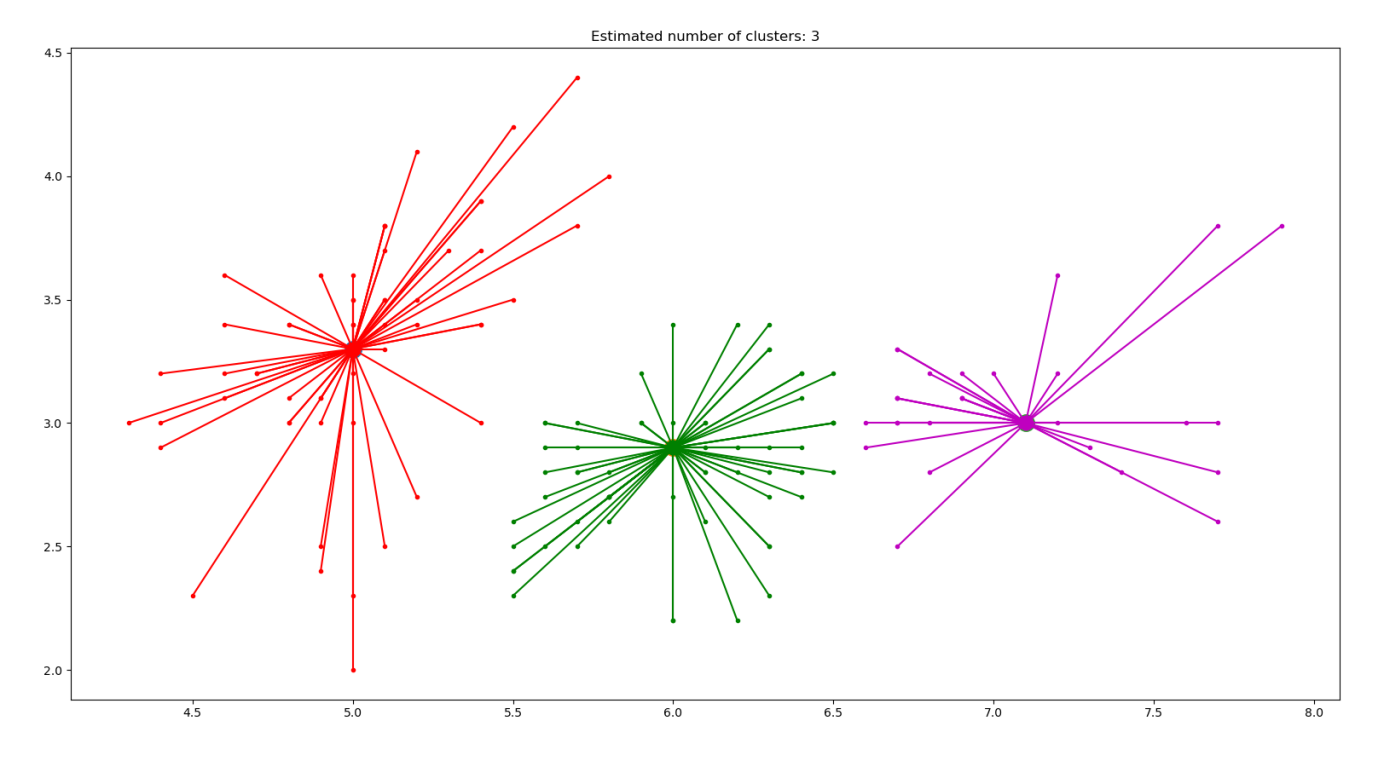


Figure : Affinity Propagation Algorithm

Out of all 3 algorithms that I used the one with the greatest level of inaccuracy was the Affinity Propagation by far, this was because it estimates how many clusters are needed, which is useful in other situations where I don’t have a definitive number of clusters needed, but as seeing as in the Iris dataset there is already 3 predetermined clusters of each species type it wouldn’t make a great deal of sense to actually use this algorithm. The K-Means algorithm has the lowest level of inaccuracy out of the 3 algorithms as it did perfectly predict one cluster right, however due to the overlapping points it meant that it wouldn’t be perfect with the other two clusters.

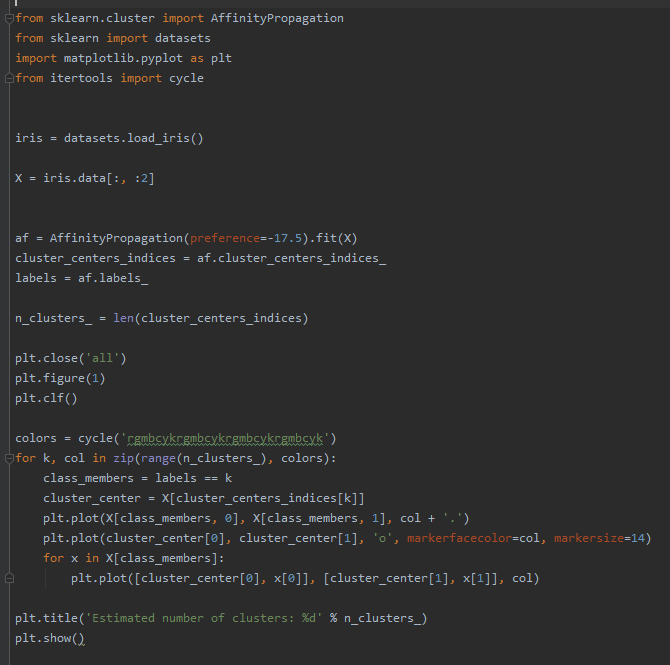
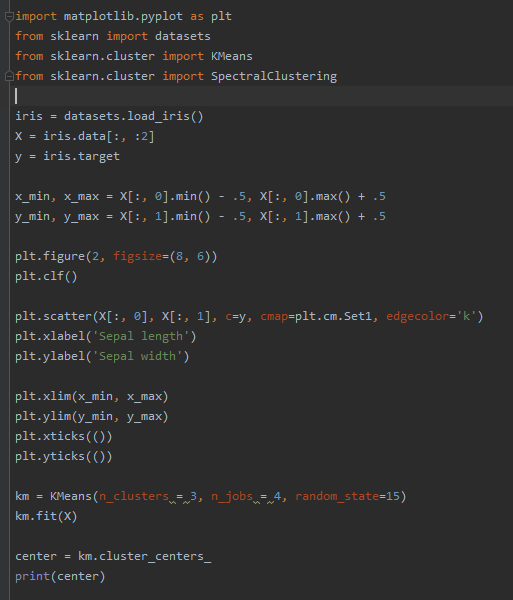


Figure : Python code for K-Means and Spectral Clustering

In Figure 5 it shows my python code that used to apply the K-Means and the Spectral Clustering algorithms to the Iris dataset, with the majority of the code is for the K-Means algorithm and the plotting of the original iris dataset into a graph.

This line is implementing the algorithm to the datasets with n\_clusters being the K-Value

This entire block of code is turning the results of the algorithm into a graph with the original Iris dataset graph alongside it, as seen in figure 2. To do this I had to set the labels of each axis for both graphs as well as the titles of the two graphs, I had the two graphs side by side so that it would be easier to compare them. The line plt.show() is the line of code needed to actually be able to see the graph once the program has been run.

This block of code is for the Spectral Clustering algorithm which states the amount of clusters you want, the

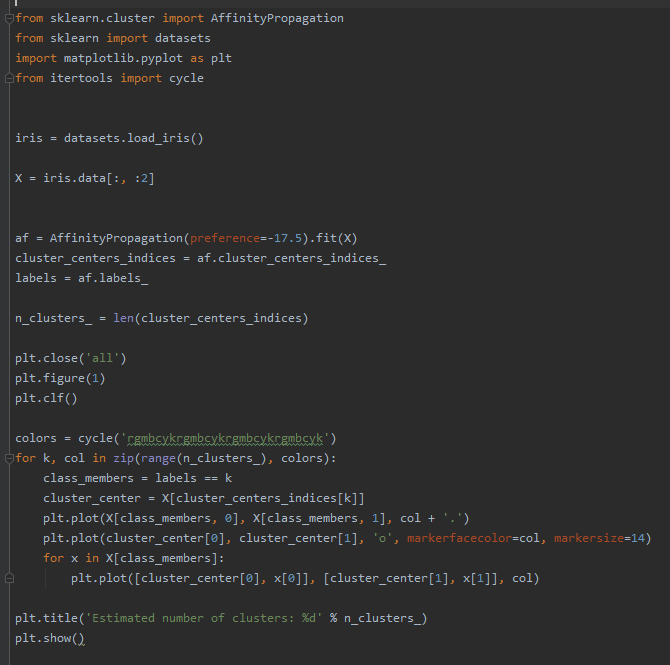


Figure : Code for Affinity Propagation

This code is the beginning of my Affinity Propagation algorithm creating it as a variable and setting the preference, I use the preference of -17.5 as that gave me 3 clusters. The rest of the code is for the plotting of the graph

I used a few different APIs when I was writing my code, with SciKit learn and Matplotlibs being the main two that I used, I also used itertools to be able to use the cycle function for the colours of the Affinity Propagation graph.

The SciKit learn API is what I used to carry all 3 of my algorithms as well as my dataset. I could’ve used MLExtend instead of SciKit learn, the reason why I didn’t use it is because I have more experience with SciKit learn and I also perceive it to be a fair amount easier to actually use.

Matplotlibs is what allowed me to plot the graphs; it has a great amount of diversity with what type of graph you require, as well as being relatively easy for you to implement them. Matplotlibs also has a wide range of colours and styles which allowed me to have the different variations of graphs.

There are other APIs which I had used in my process such as Numpy, Pandas and Seaborn. Numpy is a mathematical extension for Python, Pandas allows you to import and implement your datasets when you have them in a .csv file, and Seaborn is a Matplotlibs extension which greatly improves the graphing options. However I ended up not using them for my final iteration of files.

All 3 of my algorithms performed very well, with them all being completed extremely quickly with the execution being almost instantaneous with a high efficiency. All my results came out just as I’d expect them to with the exception of the Affinity Propagation as I didn’t expect it to have such a high inaccuracy, however the K-Means worked as I thought due to the clusters being created by using points that are in close proximity to each other, this meant that I could see before executing the algorithm roughly which points would be wrongly classified. Due to the Spectral Clustering algorithm being very similar to the K-Means algorithm I expected it to result in the same as the K-Means.

My initial aims for this project was to use a completely different dataset, I had wanted to use data that related to Overwatch League Games and I’d hoped to create an artificial intelligence driven solution that would predict who would win an upcoming game, however there wasn’t any relevant or useful premade datasets available for this use meaning that I would have to create the dataset myself, manually inputting each bit of data. Although that itself isn’t a problem the sheer amount of information I would’ve needed to record would’ve exceeded 1 thousand individual bits of data making it a massively time consuming and unviable.

Having all my data represented in the graphs makes it easy to see how it is not suited to clustering algorithms and would be better suited for other machine learning methods as my clustering algorithms are mostly based on relative distance points are from each other. I could’ve also implemented more parts of the Iris dataset, in my working I have only used Sepal Length and Sepal width, and however there are two other sets of values as well Petal Width and Petal Length which could greatly improve the possibility the clustering algorithms can distinguish the original species clusters. In the future I would’ve improved on my work by introducing larger amount of clustering algorithms as well as the possibility of other algorithms like genetic algorithm.

# Bibliography

Brownlee, J. (2020, April 6). *10 Clustering Algorithms with Python*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/clustering-algorithms-with-python/

Maklin, C. (2019, July 1). *Affinity Propagation Algorithm Explained*. Retrieved from Towards Data Science: https://towardsdatascience.com/unsupervised-machine-learning-affinity-propagation-algorithm-explained-d1fef85f22c8

SciKit Learn. (n.d.). *Affinity Propagation*. Retrieved from SciKit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AffinityPropagation.html

SciKit Learn. (n.d.). *The Iris Dataset*. Retrieved from SciKit Learn: https://scikit-learn.org/stable/auto\_examples/datasets/plot\_iris\_dataset.html

*Stack Overflow*. (n.d.). Retrieved from Stack Overflow: https://stackoverflow.com/

VanderPlas, J. (n.d.). *Python Data Science Handbook - In Depth: k-Means Clustering*. Retrieved from Github: https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html