Fundamentals of Data Structures and Algorithms for AI

Clustering Algorithms

Prepared By

Sintayew Zekarias and Fikir Awoke

December 04, 2021

Course Instructor - Dr. Beakal Gizachew

Addis ababa Institute of Technology

School of Information Technology and Engineering

Table of Contents

1. Introduction

Clustering refers to dividing data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups. For this clustering or grouping to take place, clustering algorithms play a significant role. Clustering Algorithms are categorized under unsupervised learning algorithms, where the input is not labeled and problem solving is based on the experience that the algorithm gains from solving similar problems. There are plenty of clustering algorithms that follow a different set of rules for defining the similarity among data points. The aim of this project is to implement K-means clustering and hierarchical clustering algorithms on the iris data set and predict the type of flower in an unsupervised manner. Both K-means and hierarchical clustering algorithms are considered as methods of cluster analysis, where K-means uses a pre-specified number of clusters, and hierarchical clustering builds a hierarchy of clusters without having a fixed number of clusters. The purpose of the report is to show how the implementation of the K-means and hierarchical clustering algorithms on the iris dataset went and to illustrate all the necessary methods used throughout the implementation in order to make the clustering and prediction of the Iris flowers.

2. Methodology

The following methodologies are used to achieve the objective of the project.

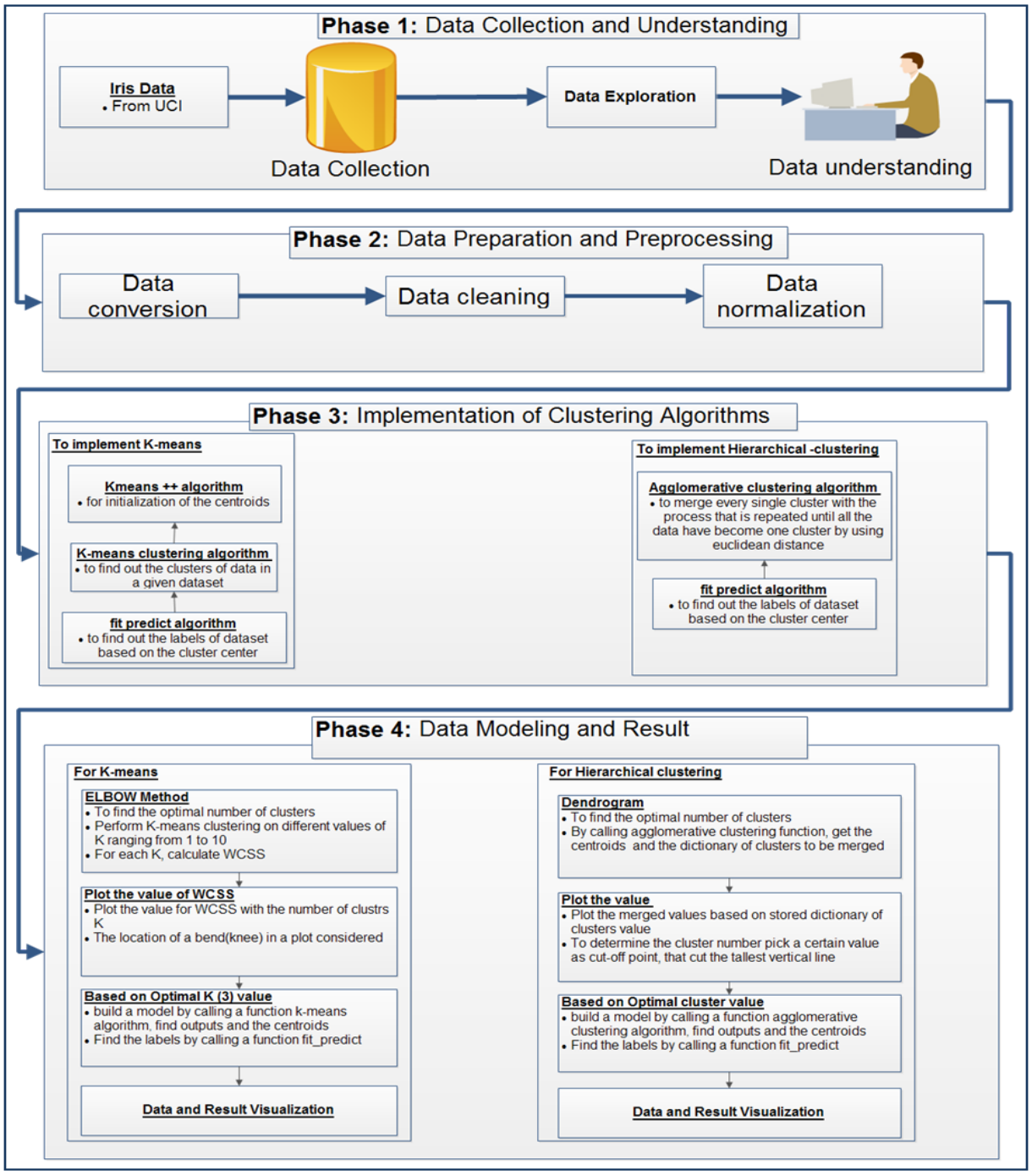

Figure 1: Methodology

2.1. Data Collection and Exploration

We collected the Iris Dataset from the University of California Irvine (UCI), Machine Learning Repository. For the sake of readability we converted the data into a csv file and began the exploration. While exploring the iris dataset, we tried to check out how the target variables are categorized, the correlation of the data and the data distribution before the clustering algorithms are applied.

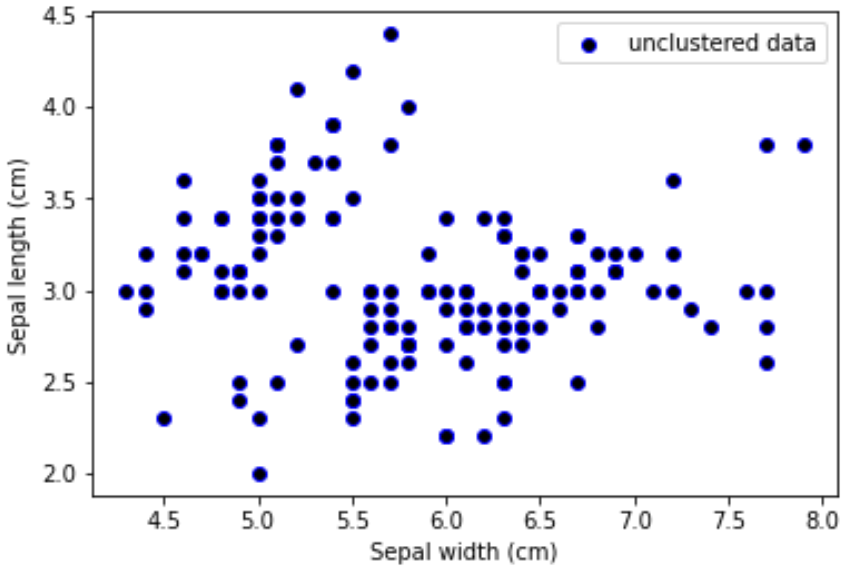

Figure 2: Unclustered Data

2.2. Data Preparation and Preprocessing

For the clustering to be made in an unsupervised learning environment, we need to remove the target variable, therefore first we converted the 'target' feature which holds the flower type to a numerical data and stored the labels into a separate variable. After that we removed the 'target' feature from the dataset. Then we normalized the dataset on the pre-processing step. We used the *iloc* function to get the features we require and the *values* function to get an array of the dataset while preparing our data for the clustering algorithms. We also assigned the number of training data to ‘m’ and number of features to ‘n’ and chose the number of iterations that might guarantee convergence.

2.3. Implementation of Clustering Algorithms

2.3.1. K-Means Clustering

K-means is the most popular unsupervised machine learning algorithm used to find out the clusters of a dataset by iteratively partitioning the given data into various clusters (groups). K refers to the total number of clusters to be defined in the entire dataset.

K-means ++ initialization algorithm

Before we go straight to the implementation process of K-means, we have to follow an effective approach when selecting centroids as it is an essential step in implementing the K-means algorithm. In a random initialization of centroids, two initial centroids might appear to be very near, which will maximize the number of iterations for the algorithm to converge. Hence, we brought the K-means ++ algorithm to make sure that initial centroids become far apart from each other. The following steps briefly explain how we implemented this algorithm:

Steps involved in K-Means++ initialization:

1. Randomly select the first cluster center from the data points and append it to the centroid matrix.

2. Loop over the number of Centroids that need to be chosen (K):

3. For each data point, calculate the euclidean distance square from already chosen centroids and append the minimum distance to a Distance array.

4. Calculate the probabilities of choosing the particular data point as the next centroid by dividing the Distance array elements with the sum of Distance arrays.

5. Calculate the cumulative probability distribution from this Probability distribution. We know that the cumulative probability distribution ranges from 0 to 1.

6. Select a random number between 0 to 1, get the index (i) of the cumulative probability distribution which is just greater than the chosen random number and assign the data point corresponding to the selected index (i).

7. Repeat the process until we have a K number of cluster centers.

K-Means Clustering Algorithm

After the completion of the K-means ++ initialization algorithm, we continued our clustering algorithm K-means. We followed the following 5 steps while implementing the K-means:

Steps involved in K-Means Clustering implementation:

1. By calling the above K-Means++ initialization function to initialize randomly the cluster centers of each cluster from the data points.

2. For each data point, compute the euclidean distance from all the centroids and assign the cluster based on the minimal distance to all the centroids.

3. Adjust the centroid of each cluster by taking the average of all the data points which belong to that cluster on the basis of the computations performed in step 2.

4. Repeat the Steps 2 to 3 till clusters are well separated or convergence is achieved.

Fit predict algorithm

After we successfully applied the K-means algorithm, we then predicted the cluster index for each sample and also used to find out the labels of the dataset by calculating the euclidean distance based on the cluster center.

2.3.2. Hierarchical Clustering

Hierarchical Clustering, as the name suggests, is an algorithm that builds a hierarchy of clusters. It combines similar items into one cluster and continues to merge similar clusters into the same cluster until a single cluster is left. Hierarchical clustering uses agglomerative or divisive techniques.

Agglomerative Clustering uses a Bottom-Up approach to form clusters. That means, it starts from individual points by considering each data point as a single cluster, and then it clusters the closer data points into one. The same process repeats until it gets one single cluster.

Divisive Clustering is the exact opposite of Agglomerative clustering. It uses a Top-Down approach to form clusters. That means, it starts from one single cluster and continues to split the furthest clusters into separate clusters. The same process repeats until each data point has its own individual cluster.

Dendrogram algorithm

Before we go straight to the implementation process of hierarchical clustering,we used the dendrogram to help us decide the number of clusters. Dendrogram is a diagram that shows the hierarchical relationship between objects. The main use of a dendrogram is to work out the best way to allocate the appropriate number of clusters for the dataset. The dendrogram distance is the distance between two clusters when they combine. It determines if two or more clusters are joined together to form a single cluster.

We have used the Agglomerative Hierarchical Clustering Algorithms for our implementation and the following are the steps we followed:

1. Each data point is assigned as a single cluster

2. Determine the distance measurement and calculate the distance matrix

3. Determine the linkage criteria to merge the clusters

4. Update the distance matrix

5. Repeat the process until every data point becomes one cluster

We used one of the most common distance measurements called Euclidean Distance to calculate the distance between two data points. And for calculating the distance between two clusters, there are about four methods named Closet Points, Furthest Points, Average Distance, and Distance between Centroids. Among these, we used the Average Distance.

This method takes the average distance of all the data points and uses this average distance as the distance of two clusters. It is known as Average-linkage.

$$d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Euclidean Distance Formula

$$L(r,s) = \frac{1}{n_r n_s}\sum_{i=1}^{n_r}\sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

Average linkage formula

Agglomerative fit predict algorithm

After we successfully applied the dendrogram algorithm, we then predicted the cluster index for each sample and also used to find out the labels of the dataset by calculating the euclidean distance based on the cutoff point.

2.4. Model Building

K-means clustering

In the K-means clustering algorithm we applied the ELBOW Method to find the appropriate number of clusters. The elbow method is one of the most popular methods to determine the optimal value of k. It uses the sum of squared distances (SSE) of every data point from its corresponding cluster centroid, which is called WCSS (Within-Cluster Sums of Squares).

The steps we followed in carrying out the ELBOW Method:

1. Perform K means clustering on different values of K ranging from 1 to any upper limit. NB : we took the upper limit as 10.

2. For each K, calculate WCSS

3. Plot the value for WCSS with the number of clusters K.

4. The location of a bend (knee) in the plot is considered as an indicator of the appropriate number of clusters or the point after which WCSS doesn't decrease more rapidly is the appropriate value of K.

Hierarchical clustering

In the Hierarchical clustering algorithm we applied the Dendrogram to determine the optimal number of clusters, and it selects the optimal value of cluster number to be 3 based on the highest vertical distance that doesn't intersect with any clusters is the middle one. To fit hierarchical clustering to the iris dataset we used Agglomerative Hierarchical Clustering. We can use a dendrogram to visualize the history of groupings and figure out the optimal number of clusters.

1. Determine the largest vertical distance that doesn't intersect any of the other clusters
2. Draw a horizontal line at both extremities
3. The optimal number of clusters is equal to the number of vertical lines going through the horizontal line

3. Results

3.1. K-Means Clustering Result

As shown on the graph below, it is clearly seen that the optimum cluster is where the elbow occurs. This is when the within-cluster sum of squares (WCSS) doesn't decrease significantly with every iteration. The graph shows that there is no bend after the 3rd cluster and also no quick decrease in the WCSS is seen, hence, 3 is the optimum number of clusters, i.e, k=3.

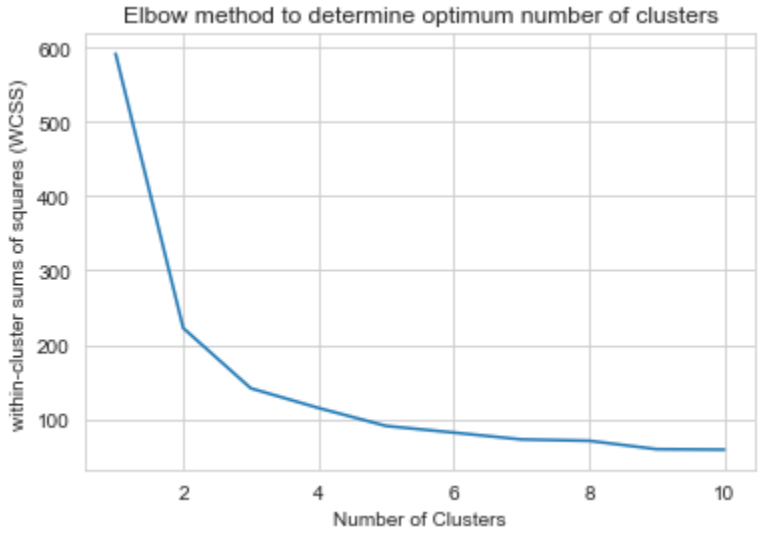

Figure 3: Elbow Method

We have also implemented all these steps using the sklearn library so that we can compare the results with our own result, based on that the following graph shows the Elbow method to determine the optimal number of clusters, and it selects the optimal value of k to be 3 based on the bend.

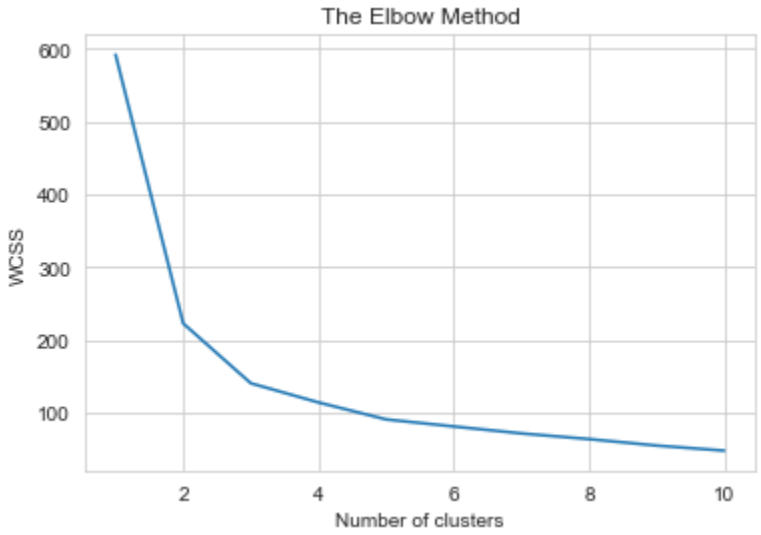

Figure 4: Sklearn Library Elbow Method

After Applying the K-means to our dataset, the results of the clusters are plotted on the following graph.

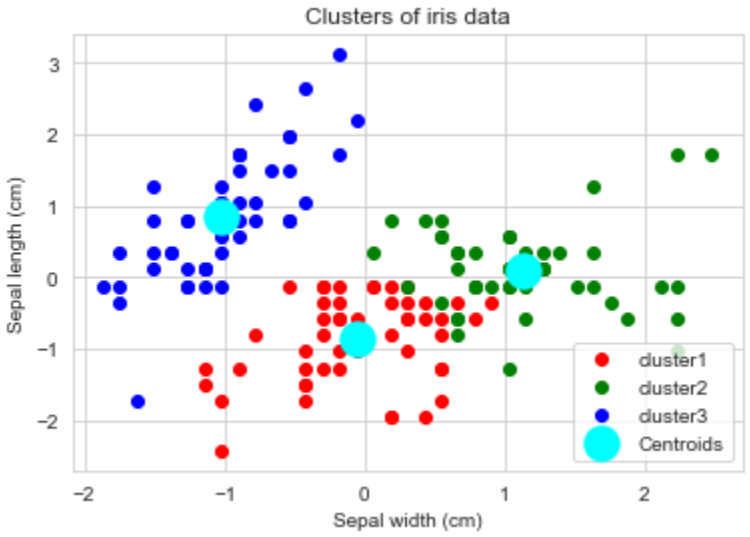

Figure 5: Clusters of Iris Dataset

And based on the sklearn library implementation the cluster looks like this:

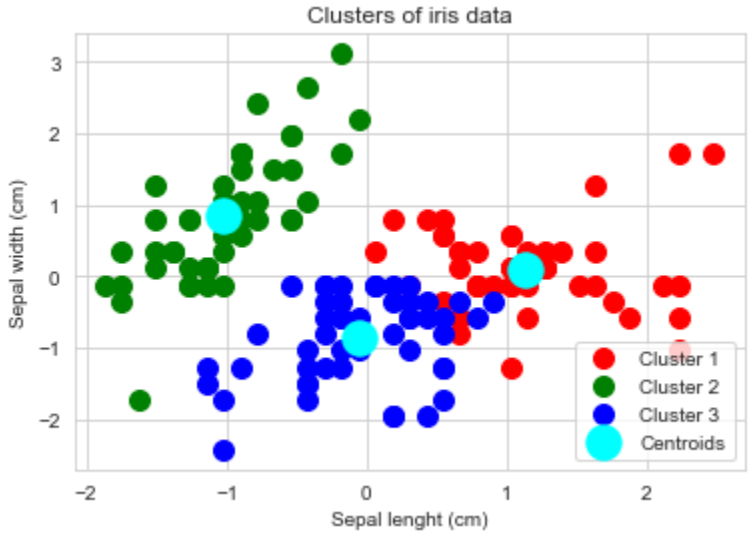

Figure 6: Sklearn Library Clusters of Iris Dataset

While Comparing the Actual and the Predicted values, we got the following cluster distribution result.

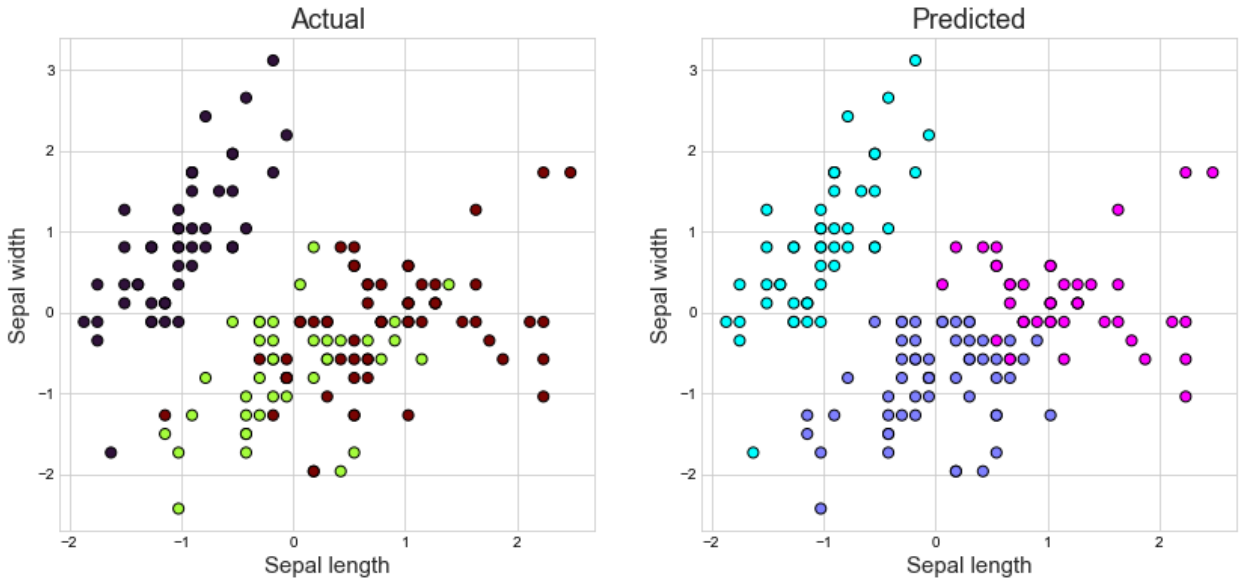

Figure 7: Actual and Predicted clusters

And this is the actual and predicted values for the clustering made using the sklearn library.

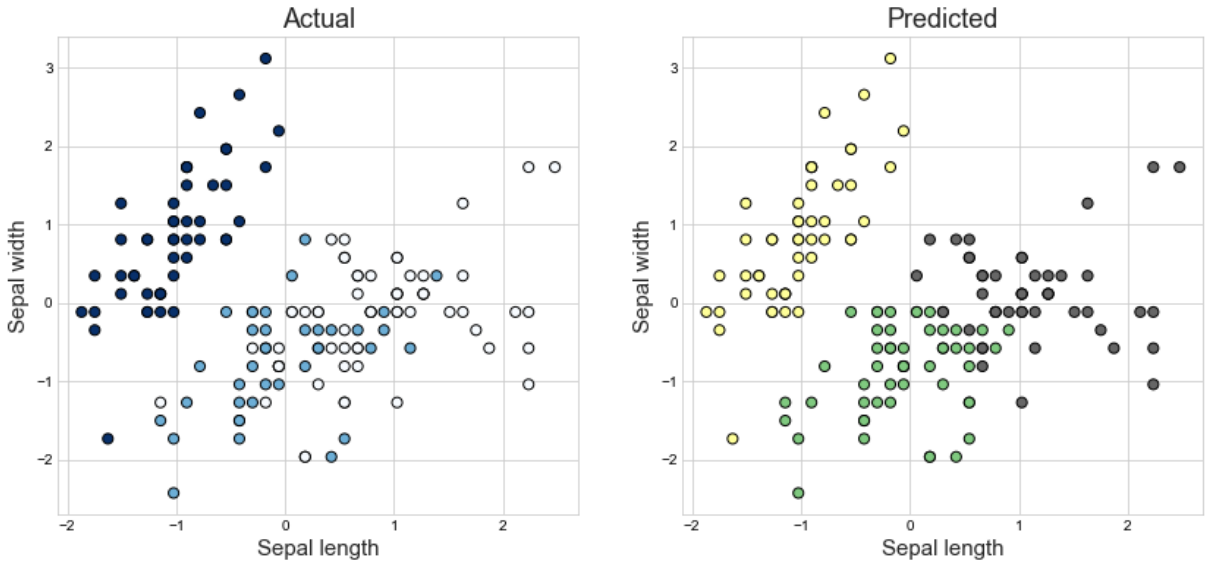

Figure 8: Actual and Predicted clusters in sklearn library

3.2. Hierarchical Clustering Result

In the Hierarchical Clustering Algorithm the result of the dendrogram on the iris datasets looks like this when plotted on a graph :

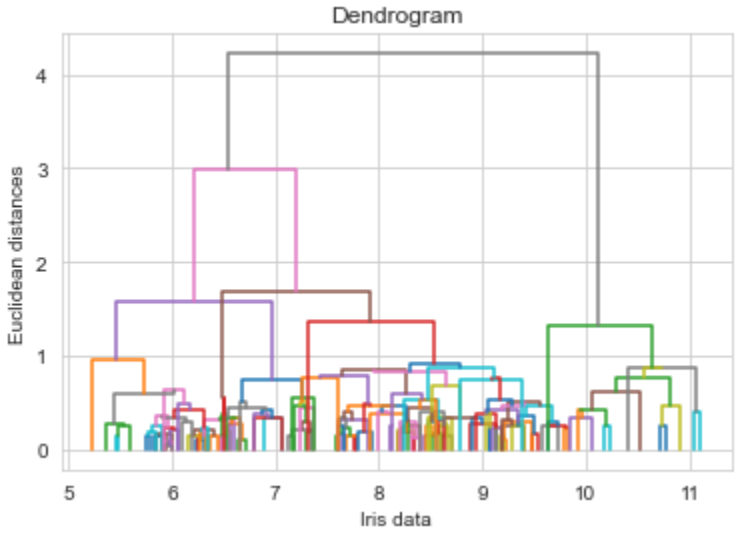

Figure 9: Dendrogram

We have also implemented all these steps using the scipy.cluster.hierarchy library so that we can compare the results with our own result, based on that the following graph shows the Dendrogram to determine the optimal number of clusters, and it selects the optimal value of cluster number to be 3 based on the highest vertical distance that doesn't intersect with any clusters is the middle one.

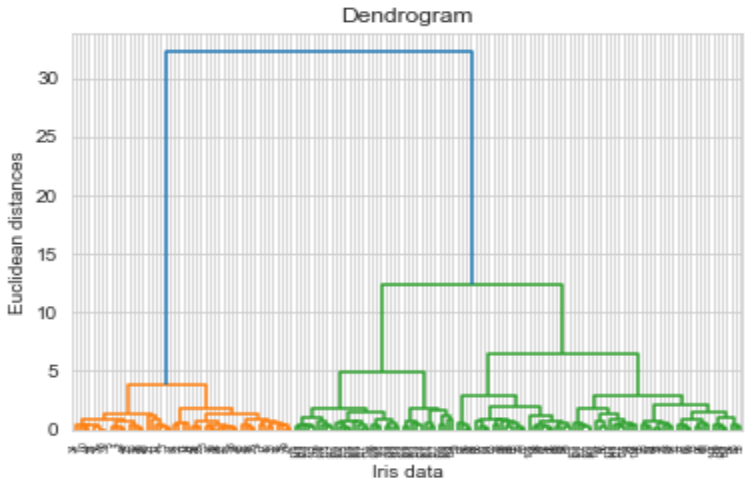

Figure 10: scipy.cluster.hierarchy library *dendrogram*

To find the Optimal Number of Clusters , we used Dendrogram.So a cut-off point is at 146 we would end up with 3 different clusters. After Applying the hierarchical clustering to our dataset, the results of the clusters are plotted on the following graph.

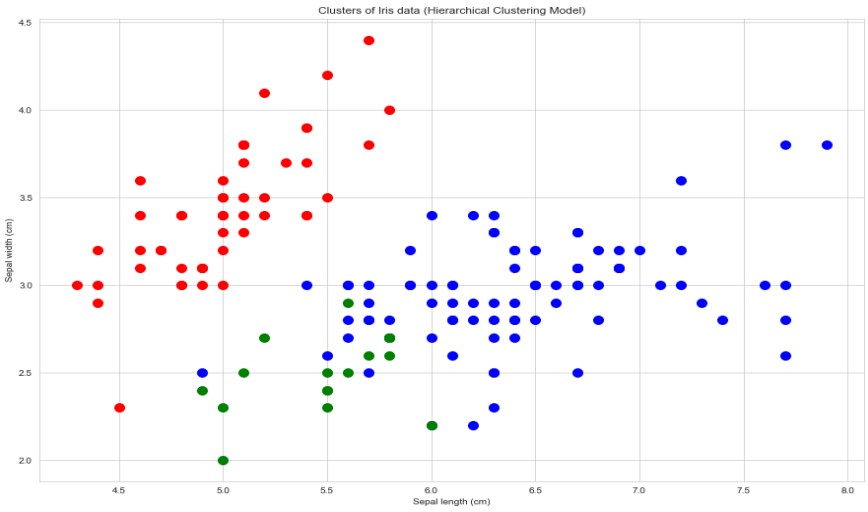

Figure 11: Clusters of Iris Dataset

And based on the sklearn library implementation the cluster looks like this:

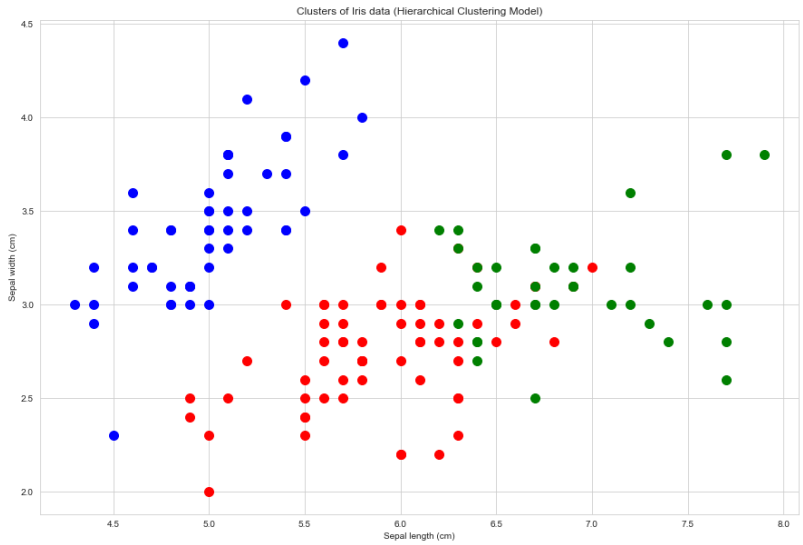

Figure 12: Sklearn AgglomerativeClustering Library Clusters of Iris Dataset

4. Conclusion

We have implemented the K-means and Hierarchical Clustering algorithms from scratch and we tried to compare our results with that of the built in library and showed the optimal value. We have learned a lot throughout the implementation process.

References

https://archive.ics.uci.edu/ml/datasets/iris

https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/?utm_source=blog&utm_medium=beginners-guide-hierarchical-clustering

https://www.geeksforgeeks.org/difference-between-k-means-and-hierarchical-clustering/

https://medium.com/machine-learning-algorithms-from-scratch/k-means-clustering-from-scratch-in-python-1675d38eee42

https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering/

https://www.mltut.com/hierarchical-clustering-in-python-step-by-step-complete-guide/