**CS 408 - Group 3**

**Project Report 04 - Implementing K-means Algorithm**

**1.Introduction**

K Means algorithm is an iterative algorithm that tries to partition the dataset into ‘k’ number of pre-defined, distinct, non-overlapping subgroups (clusters) where each data point belongs to only one group. Each data point is assigned to one cluster, aiming to minimize the sum of squared distances between data points and their respective cluster centroids. The algorithm strives to make intra-cluster data points as similar as possible.Each cluster's centroid is the arithmetic mean of all data points in that cluster.The goal is to minimize the variation within clusters, resulting in more homogeneous clusters.

Other than that compared to the other clustering methods such as mean-shift this method is less computationally expensive in any mean.But as the downfall of this algorithm k should be pre-determined and that needs the expertise support and depends on the initialization of the centroid point.As for the relief in lot of cases it always converges to the same result.

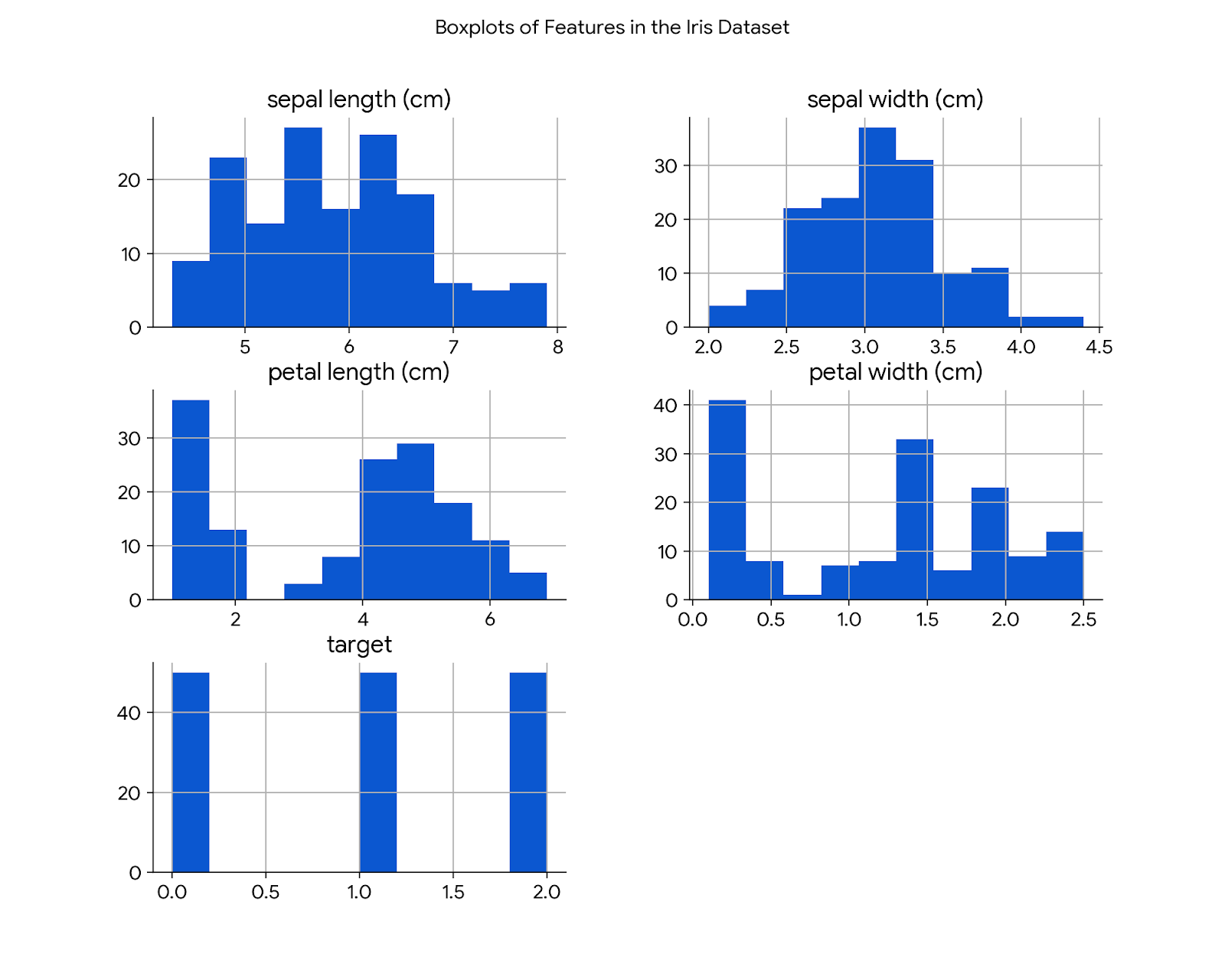
**2.Data set**

Iris dataset [1] was introduced by the British statistician and biologist Ronald A. Fisher in his 1936 paper titled "The Use of Multiple Measurements in Taxonomic Problems."[2] It is one of the most famous datasets in the field of machine learning and statistics. It consists of 150 samples from three species of Iris flowers Iris setosa, Iris versicolor, and Iris virginica(Figure 2.1). The data set has 4 feature columns such as sepal length,sepal width,petal length and petal width in centimeters.



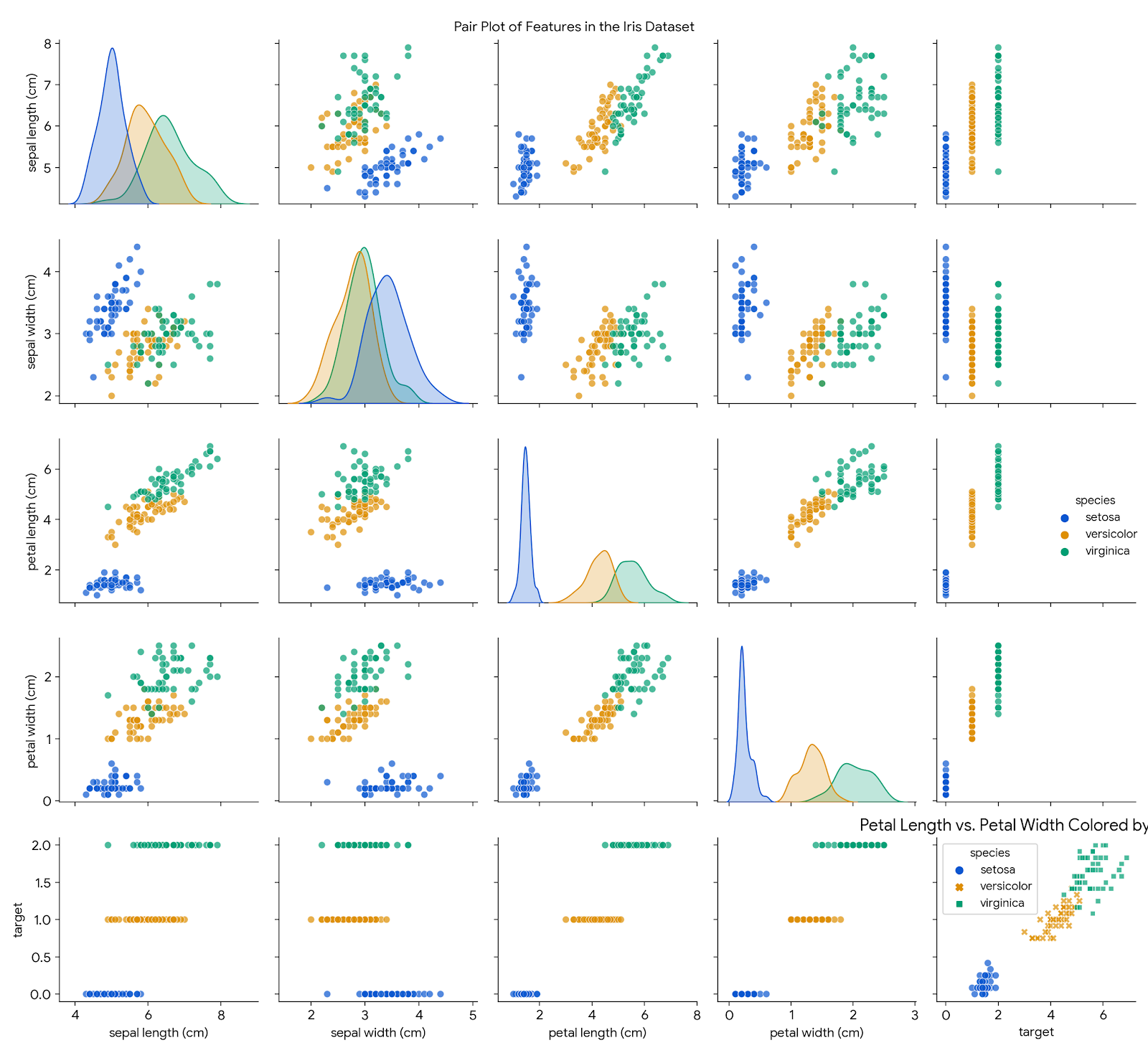
**Figure 2.1** : *iris setosa, iris versicolor , iris virginica images*

As by analyzing the data set graphically we can see the connection between the according features with Exploratory data analysis(EDA) on the dataset[3].



**Figure 2.2** : *Histograms of features in the iris dataset*

By analyzing these Histograms(Figure 2,2) we can clearly understand the features such as sepal length and width, show overlapping distributions for different species, indicating potential challenges in using these features alone for classification.Petal length and width shows more distinct distributions for the three species of iris flowers and this is very usable in an classification problem.

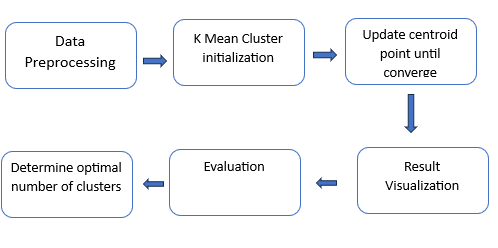


**Figure 2.3** : *Pair plots of features in the iris dataset*

With analyzing pair plots(Figure 2.3) on the dataset we can identify that the relationship between two features and each color represent the different type of the species.The plot clearly shows that petal length and petal width are the most discriminative features for separating the three Iris species. There's a distinct separation between Iris setosa and the other two species based on these features.And we can identify that there is an overlap between Iris versicolor and Iris virginica while plotting sepal length and width.

The key points of this analysis shows that petal length and petal width are the most promising features to be included in the clustering while the overlapping on And data normalization should be added to balance the consideration of all features equally.

**3.Methodology**

****

**Figure 3.1 :***Methodology*

* Data preprocessing - This is the procedure which includes analyzing the data set using the EDA (Exploratory data analysis) which gives the better idea about the dataset.We have explained this part under dataset.
* K mean clustering
* Initialization - In this part the libraries should be initialized and then the initial k value should be assigned.
* Assign centroid point - By this part the random starting point should be assigned to the
* Update centroid - update the placement of the centroid by correcting it with distance
* Iteration - we should run the algorithm until we get the convergence of the points in the cluster to the centroid
* Evaluation - evaluate the correctness of the clustering with the silhouette coefficient analysis
* Determine optimal number of clusters - After the evaluation we can conclude our decision on the optimal number of clusters that we should use in order to get the best results.

**4.Implementation**

For the implementation of K- mean cluster with python we can use the libraries such as numpy,pandas,matplotlib,seaborn.sns and sklearn.cluster

In the building of the code from scratch without using inbuilt functions For that we can use the random and math libraries as addition.

4.1 Distance Calculation

Then as the first step we have to decide on what is the distance type that we have to use to get the result as well known we can use the distance type such as Euclidean Distance , Manhattan Distance ,city block distance and hamming distance.But rather than using another method we stick to use the euclidean method because it is easy and efficient to implement it gives the geometrically calculated distance measures.

But if the dataset has not been in a different shape maybe Manhattan distance will be good to use.

4.2 Centroid Initialization

For initializing the centroid and the k value as the initial number of clusters we choose the K -means ++ method and as the initial number of clusters we are going to try on using k = 2,3,4 .

With using k - means ++ method since it uses the most further away point it will give the convergence of the points more quickly

def initialize\_centroids\_plusplus(data, k):

"""Initializes centroids using K-means++ algorithm."""

centroids = [data[np.random.choice(data.shape[0])]]

for \_ in range(1, k): distances = np.min([euclidean\_distance(point, centroid) for centroid in centroids], axis=0)

probabilities = distances / np.sum(distances)

next\_centroid = data[np.random.choice(data.shape[0], p=probabilities)]

.append(next\_centroid) return np.array(centroids)

We have to save them in an array in case we want to visualize them at the end of the result or use them in that process.

4.3 Cluster Assignment

In the cluster assignment calculates the distance between each data point and all centroids using np.linalg.norm and it assigns each data point to the nearest centroid using np.argmin function.With this part the dataset would be normalized as well.

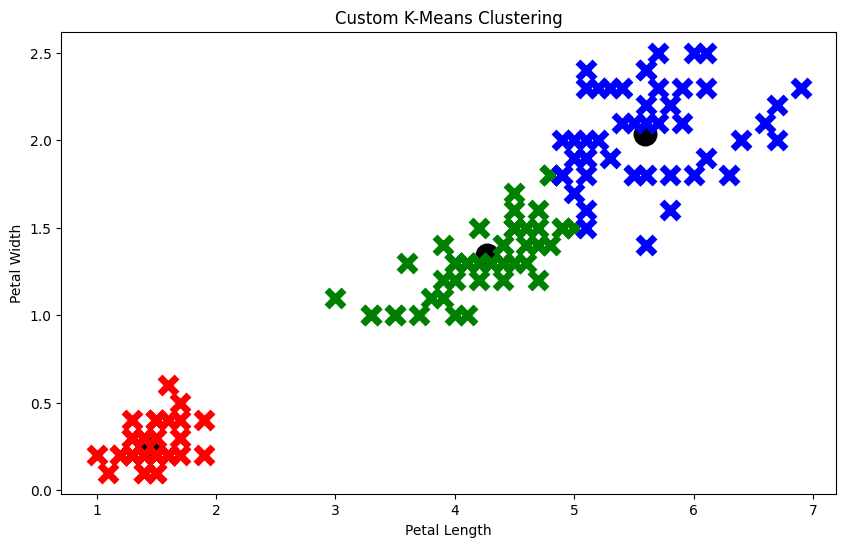
4.4 Centroid Update

Centroid calculation involves determining the mean of all data points assigned to a specific cluster. This newly computed mean then replaces the old centroid for that cluster To find the mean we use the function np.mean.

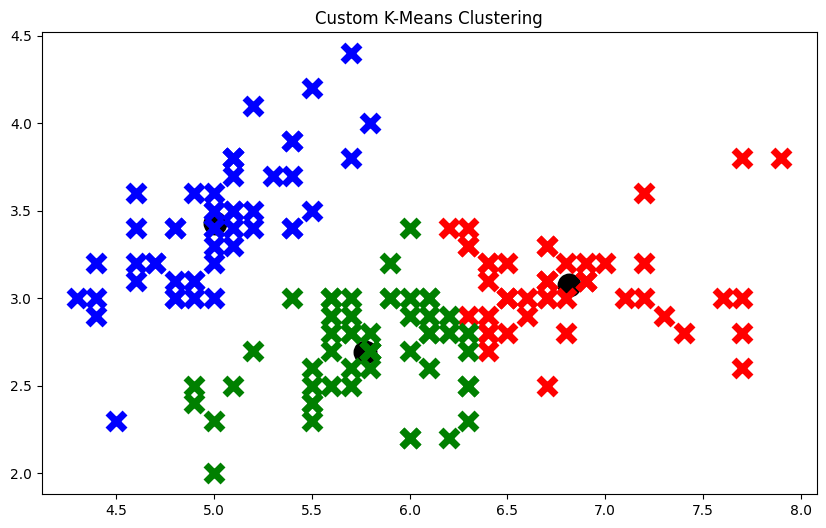
4.5 Convergence Criteria

The convergence criteria would be if all centroids haven't changed significantly across iterations using np.all(centroids == new\_centroids).In addition to that the iteration of the loop should be set without that it may lead to an infinite loop.

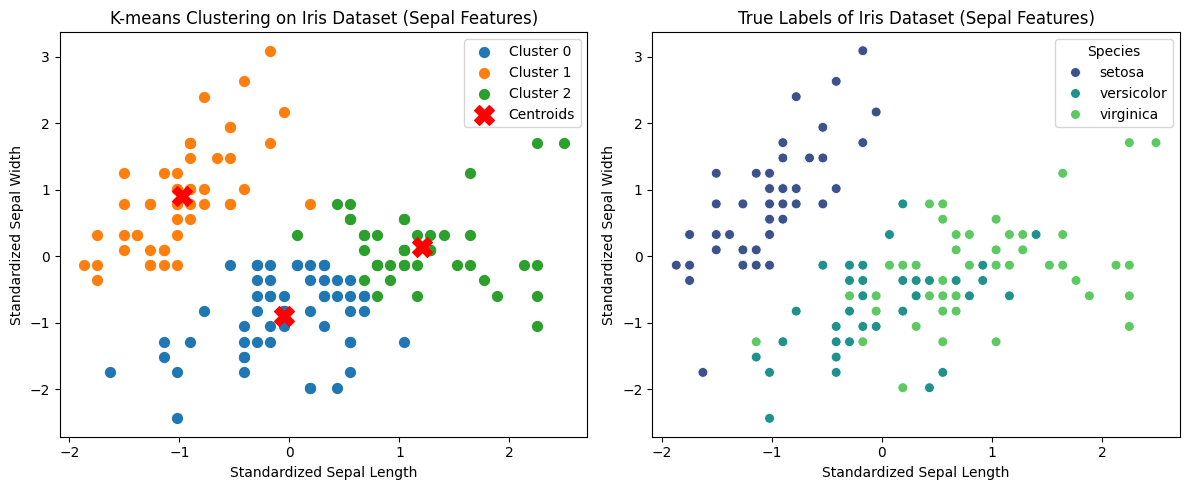
**5.Results**

****

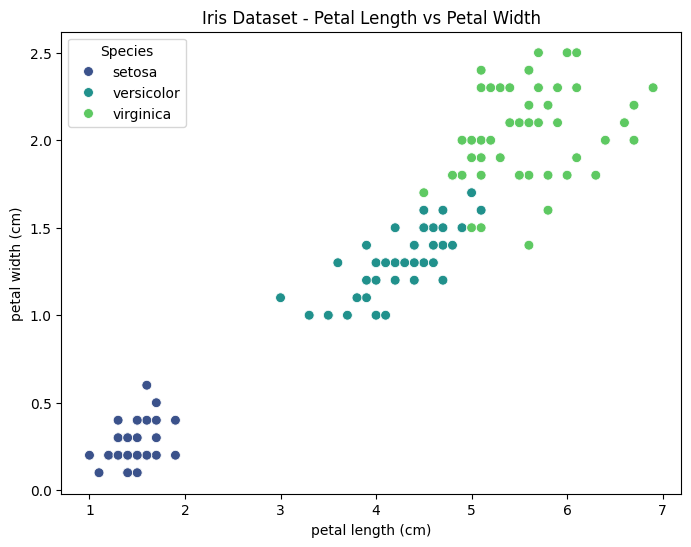
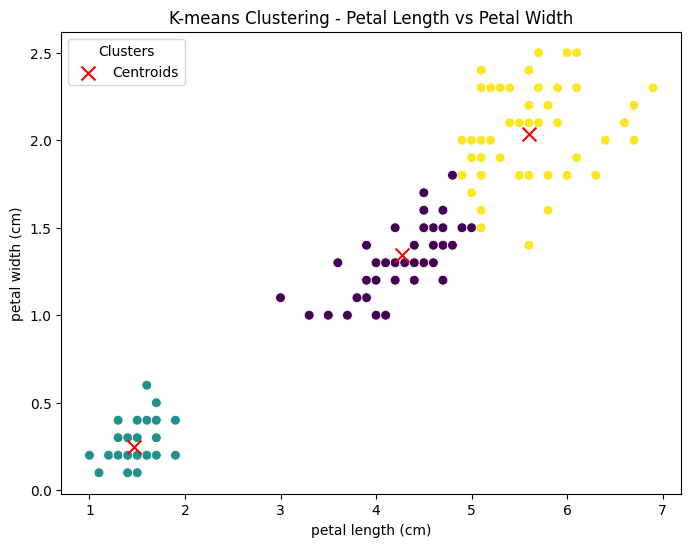
**Figure 5.1** : *Petal Length and Petal width cluster*

****

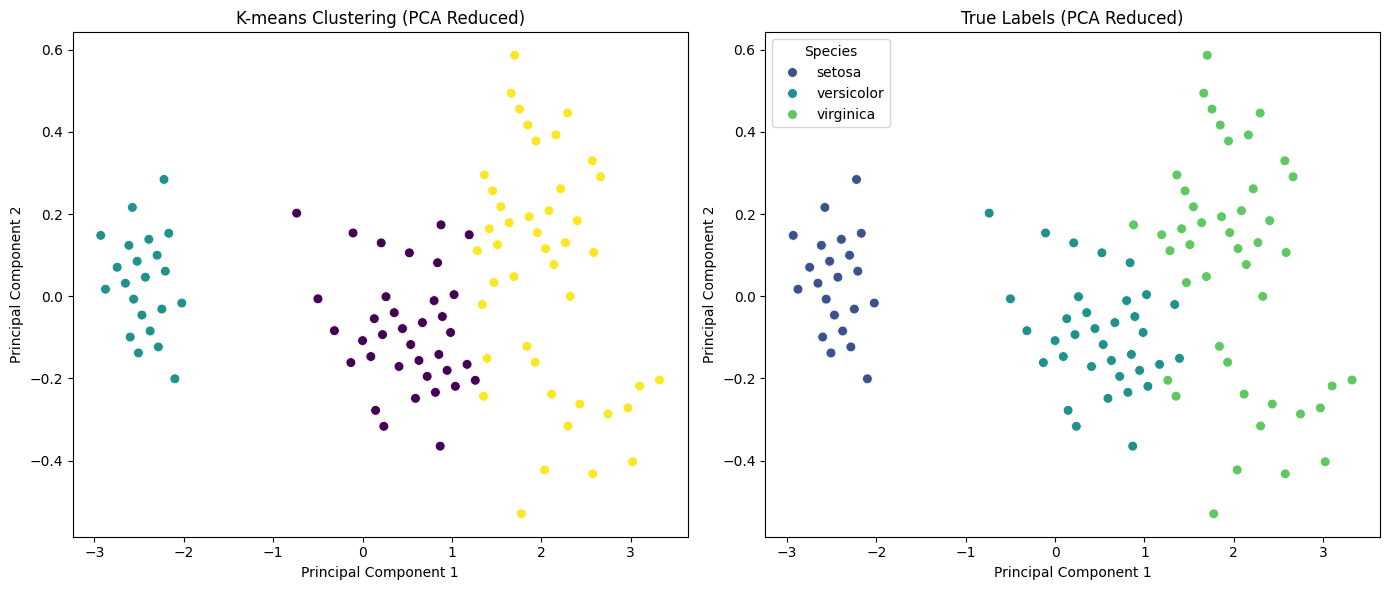
**Figure 5.2** : *Sepal Length and Sepal width cluster*

****

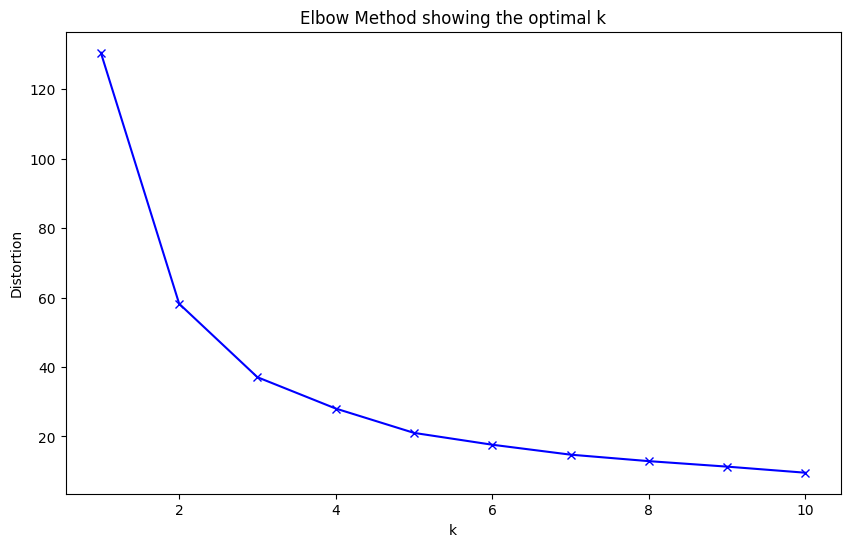
**Figure 5.3** : *Sepal Length and Sepal width cluster Comparison with true labels*

**

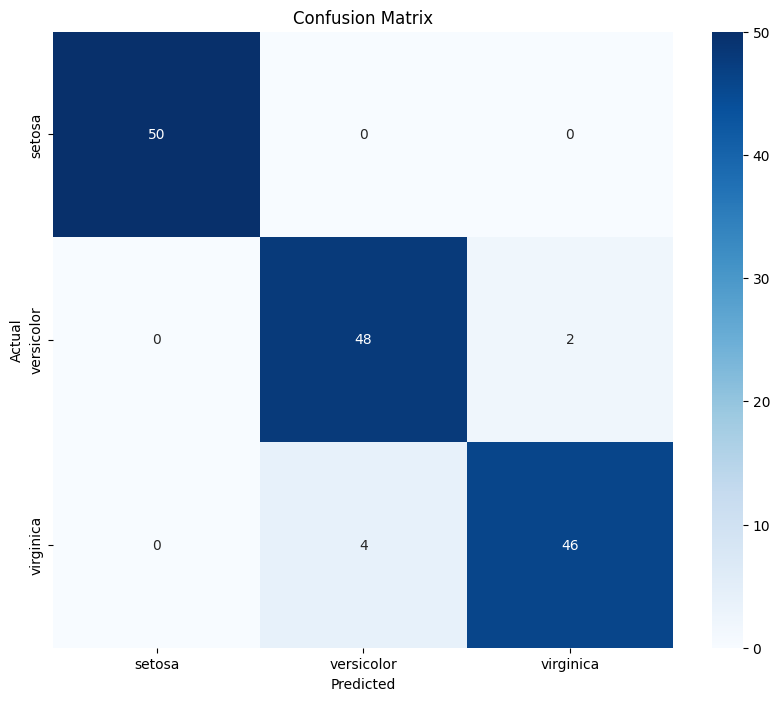
**Figure 5.4** : *Petal Length and Petal width cluster Comparison with true labels*

**

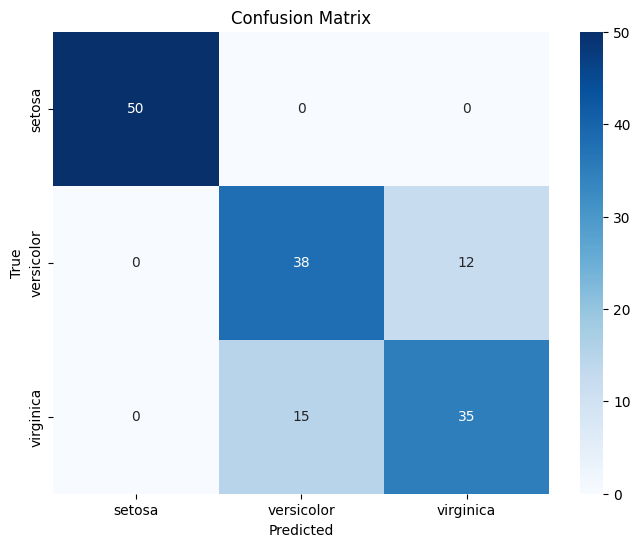
**Figure 5.5** : *K mean cluster Comparison with true labels with PCA Reduced*

****

**Figure 5.6** : *Elbow method showing the optimal k*

**

**Figure 5.6** : *Confusion matrix for petal width and length*

**

**Figure 5.6** : *Confusion matrix for sepal width and length*

**6.Discussion**

The implementation of K-means clustering to the Iris dataset shows its ability in partitioning the data into three distinct clusters, aligning closely with the predefined species. This results highlights the algorithm's ability to identify meaningful patterns within the data.

Analysis of results:

As the result we obtain by creating the graph on petal length and petal width we can clearly see that with the comparison to the true points in the input data set the outliers have been removed got the most corrected 3 clusters.on the target values as iris setosa,iris viginica and Iris versicolor.

The Elbow method (Figure 5.3) is often used to determine the optimal number of clusters by plotting the sum of squared distances from each point to its assigned cluster center. The "elbow" point in this plot indicates the optimal number of clusters. This method confirms that three clusters are appropriate for the Iris dataset, supporting the choice of k=3 in our implementation.

Limitations:

One significant limitation of K-means is its assumption that clusters are spherical and of roughly equal size. This assumption can be restrictive, particularly when dealing with datasets that have complex feature distributions or non-spherical clusters. In the Iris dataset, while the algorithm performs well, this limitation may affect the clustering outcome in more complex scenarios.

K-means is sensitive to the initial placement of centroids. Different initializations can lead to different clustering results, which may impact the final outcome. Although techniques like K-means++ can help mitigate this issue, the inherent sensitivity remains a concern.

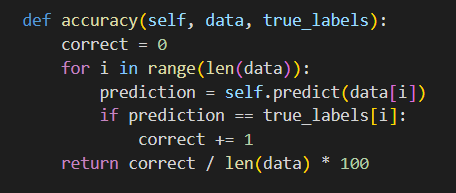
**7.Evaluation**

For evaluating the K-means clustering implementation provided above, we used several metrics.

7.1 Accuracy Calculation

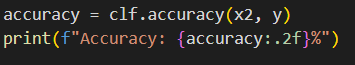
Evaluated by predicting the cluster for each data point to the true label.

Implementation:



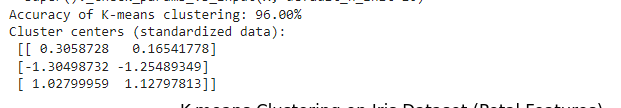
**Figure 7.1.1** : *accuracy calculation code*

Results:

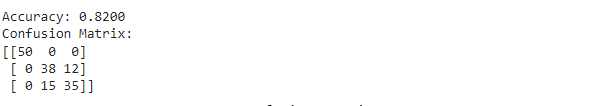


**Figure 7.1.2** : *accuracy result code*

The accuracy calculated from this implementation was 94% for petal length petal width implementation



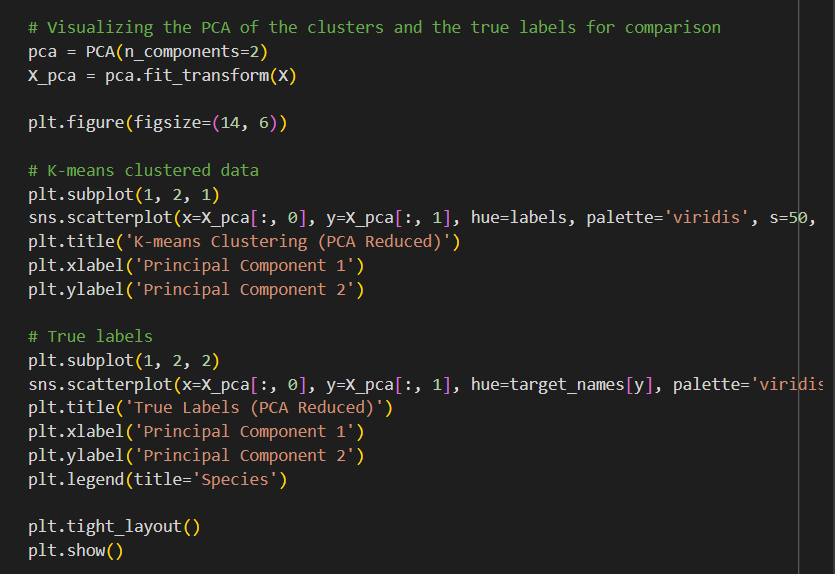
And we got 82% accuracy for the sepal length sepal width implementation



7.2 Principal Component Analysis(PCA) Visualization

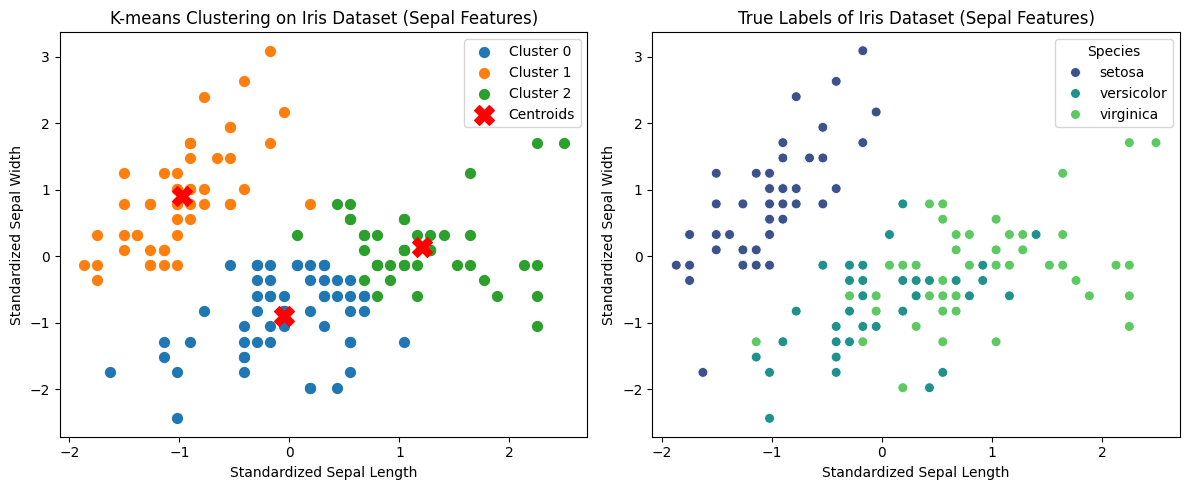
We generated two plots: one showing K-means clustering results and other showing the true labels.

Implementation:



**Figure 7.2.1**: *PCA reduction code*

Results:



**Figure 7.2.2** : *PCA reduction result*

**8.Conclusion**

In this project, we implemented the K-means algorithm to cluster the Iris dataset. We preprocessed the data, initialized the centroids using k-means ++ method, assigned data points to clusters and iteratively updated the centroids until they became convergent.

This analysis highlights the discriminative power of petal length and petal width in clustering this Iris effectively. In our results, it shows that we could cluster the Iris dataset into 3 clusters successfully, using K-means clustering, corresponding closely to the actual species.

The limitations of the K-means algorithm can be listed as follows. Its assumption of spherical, equally sized clusters and sensitivity to initial centroid placement can lead to suboptimal clustering in more complex datasets and also determining the optimal number of clusters remains a challenge without domain knowledge or supplementary validation techniques.

In conclusion, even though the K-means algorithm is an efficient algorithm for simple clustering tasks, it needs to be used in conjunction with other clustering algorithms and validation techniques. As for the future work, we will explore alternative clustering methods such as hierarchical clustering incorporating dimensionality reduction techniques to improve clustering performance on more complex datasets.

**9.Reference**

[1] “IRIS - UCI Machine Learning Repository.” https://archive.ics.uci.edu/dataset/53/iris

[2] Fisher, R.A., "The use of multiple measurements in taxonomic problems," Annals of Eugenics, vol. 7, pp. 179-188, 1936.

[3] A. K. Jain, "Data clustering: 50 years beyond K-means," Pattern Recognition Letters, vol. 31, no. 8, pp. 651-666, 2010.

**10.Group Members**

* S/18/427 - Janith Kulatunga
* S/18/030 - Supul Edirisinghe
* S/18/562 - Sanjana Wijenayaka
* S/18/335 - Sachini Dasa
* S/18/557 - Nilmi Wickramasinghe