# Assignment 1 Report

## K-Means Clustering Algorithm Comparison with Scikit Learn Library

In this project we are assigned to find clusters in a data. In the first phase of this project, I created data and tried to cluster them in three centers. Later, I changed the code structure to implement k-means algorithm up to 10 centers. I have tested my code with $k = 3\ and\ k = 7$ . In smaller $k\ values\ like\ 2\ or\ 3$ my algorithm and scikit learn algorithm works the same. Our results are identical as you can see in Figure 1 and Figure 2.

```
Kmeans k= 3  centers:
[[ 1.98410174  0.81302038]
 [ 0.97692125  4.56883102]
 [-1.71656153  2.91261167]]
Scikit learn k= 3  centers:
[[-1.71656153  2.91261167]
 [ 1.98410174  0.81302038]
 [ 0.97692125  4.56883102]]
Optimal k value for this dataset is:  3
```

*Figure 1*

```
Kmeans k= 2  centers:
[[ 1.20212561 -0.0741969 ]
 [-0.20212561  0.5741969 ]]
Scikit learn k= 2  centers:
[[-0.20212561  0.5741969 ]
 [ 1.20212561 -0.0741969 ]]
Optimal k value for this dataset is:  2
```

*Figure 2*

But when number of centers are getting bigger, my algorithm and scikit learn algorithm's centers are started to differ slightly. The reason for this in my opinion, the initialization step. While starting to implement k-means algorithm, in the first step I assigned random centers for each point. This randomness may cause this difference between two algorithms. I set the threshold for stop condition of the algorithm to $0.0001\ as\ in\ scikit\ learn$, so this parameter would not cause any difference in final center points. You can see the centers in Figure 3 and Figure 4.

```
Kmeans k= 7  centers:
[[ 2.32251185  1.38111861]
 [-1.30653356  7.83303795]
 [ 1.55031236  8.23467096]
 [ 0.80640161  4.15715989]
 [-1.74710443  2.85084596]
 [ 5.57742653  0.29793448]
 [ 9.16706182 -2.20335938]]
Scikit learn k= 7  centers:
[[-1.34722989  7.78373153]
 [-1.86295511  2.85598439]
 [ 1.48972141  8.27458058]
 [ 5.63095568  0.27183097]
 [ 0.77360533  4.02694937]
 [ 2.41031528  1.31975809]
 [ 9.16706182 -2.20335938]]
```

*Figure 3*

## Finding Best K Automatically

In this part of the project, we need to make some research on how to find best $k\ value$ for given any dataset. I found two different sites that explains the algorithm to find the best $k\ value$. There are two different algorithms to resolve this problem. First and most used one was Elbow Method (Sinha, 2021). In this method for each $k\ value$ it calculates the average distortion from the center. As the number of centers $k$ increases, number of points in a cluster will be decreased. Thus, average distortion will decrease. Therefore, the $k\ value$ where this distortion decreased the most is the elbow point. As you can see in Figure 4, Figure 5 and Figure 6 it gives correct results in my dataset.

The second approach was The Silhouette Method (Mahendru, 2019). Silhouette value can take values in range of +1 and -1. Higher values mean more points are placed in correct clusters. There are two different parameters in this algorithm. First one was the measure of similarity of the point $i$ to its own cluster. Second parameter was the measure of dissimilarity of $i$ from points in other clusters. Euclidean Distance was used while calculating these parameters.

I used Elbow Method to find best k value. I implemented the code on my own. I used a for loop to run my kmeans clustering method for a range of clusters k (in my code 1 to 10) and

for each value, we are calculating the sum of squared distances from each point to its assigned center(distortions). After this calculation I tried to find best k value by comparing consecutive slopes ratio. It gives correct result in my first dataset. You can see the results in following pictures.

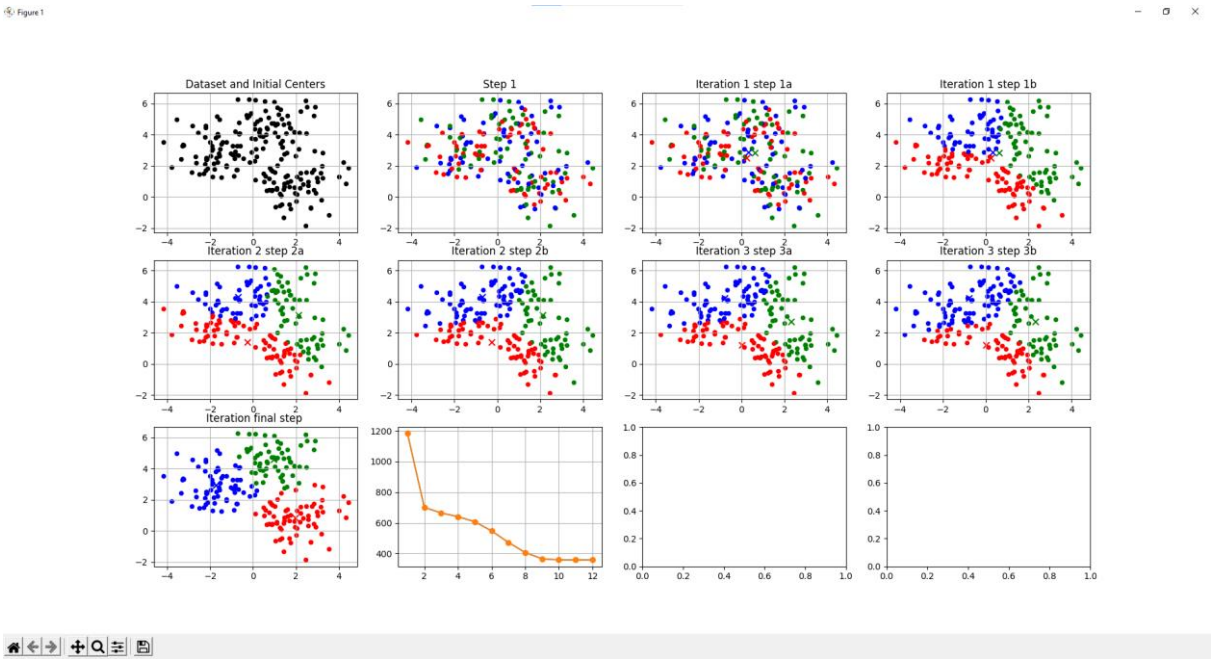


*Figure 4 My Algorithm with k=3*

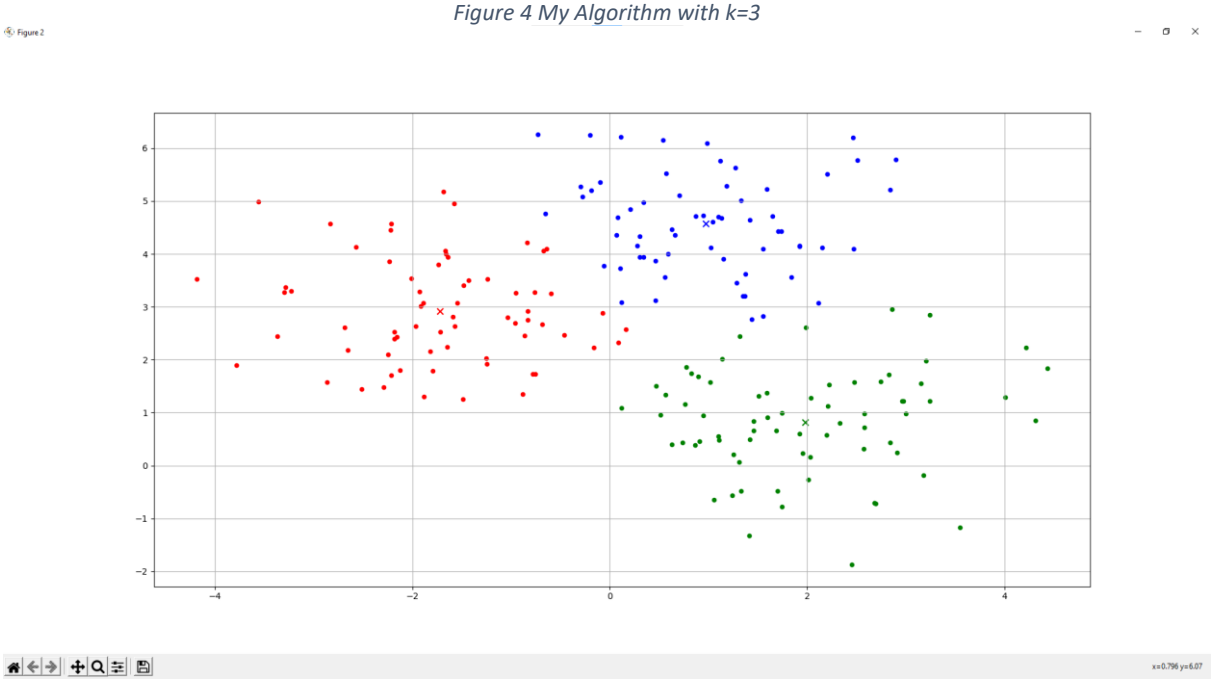


*Figure 5 Scikit Learn with k=3*

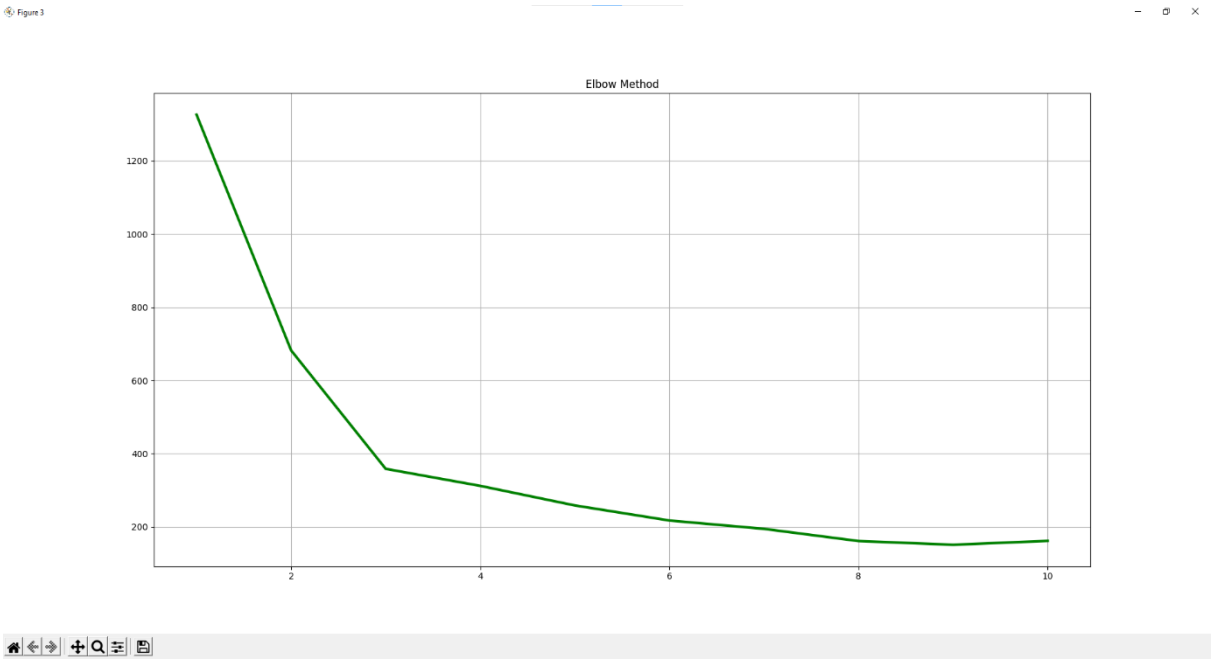


*Figure 6 Elbow Method*

# References

*GeeksforGeeks*. (n.d.). Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/ml-determine-the-optimal-value-of-k-in-k-means-clustering/

Mahendru, K. (2019, June 17). *medium*. Retrieved from medium: https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb