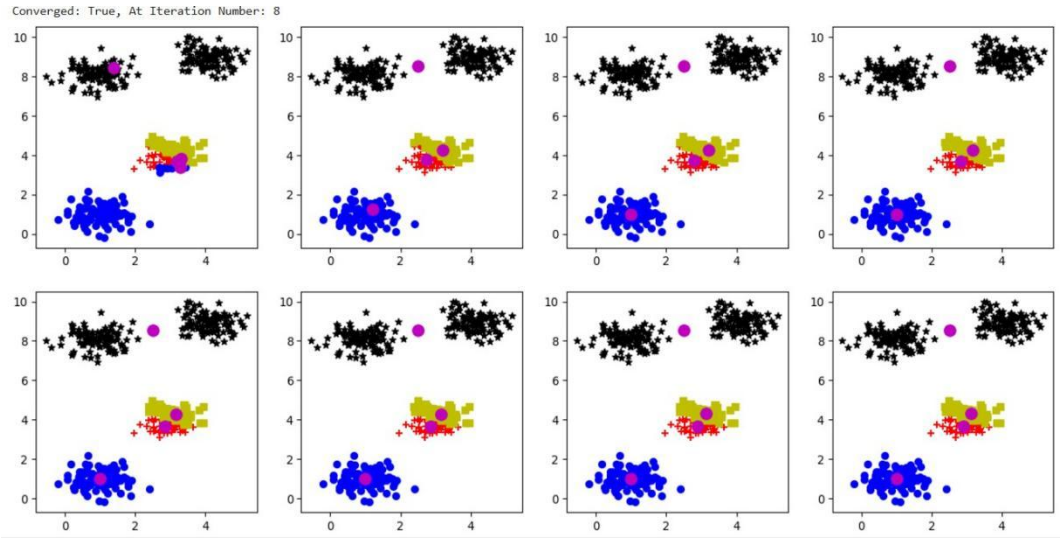
The output we got when the centroids are chosen randomly are displayed below. It took 8 iterations to converge because in the original code the centroids are initialized randomly.



Using the same data points and running the provided version of the code 10 times for 10 random choices of 4 centroids. Keeping maximum number of iterations to 16 and finding the number of iterations required by the algorithm to converge at each choice.

In this task the K-means algorithm with random initialization of the centroids is used by running it 10 runs on the same data set, each using one of the 10 different random seed values used in the code. The K-means algorithm uses data points initialized with randomly chosen centroids via np.random.permutation, and performs up to 16 iterations, ending early if centroids converge at 1e-6. For each run, the number of iterations required by the method to converge was recorded.

```
Run 1 with seed 22:
Converged: True, At Iteration Number: 10
Run 2 with seed 32:
Converged: True, At Iteration Number: 3
Run 3 with seed 42:
Converged: True, At Iteration Number: 5
Run 4 with seed 19:
Converged: True, At Iteration Number: 4
Run 5 with seed 18:
Converged: True, At Iteration Number: 12
Run 6 with seed 176:
Converged: True, At Iteration Number: 7
Run 7 with seed 24:
Converged: True, At Iteration Number: 11
Run 8 with seed 84:
Converged: True, At Iteration Number: 7
Run 9 with seed 39:
Converged: True, At Iteration Number: 5
Run 10 with seed 47:
Converged: True, At Iteration Number: 4
```

As the number of iterations required for Run 5(12 iterations) is more compared to Run 2(3 iterations). Because in run 5 the centroids were not positioned properly, leading to slower adjustments, and requiring more iterations to converge. However, in run 2 the initial centroids are positioned properly which results in quick convergence. Finally, it shows how the randomness of centroid initially can affect the efficiency of the algorithm.

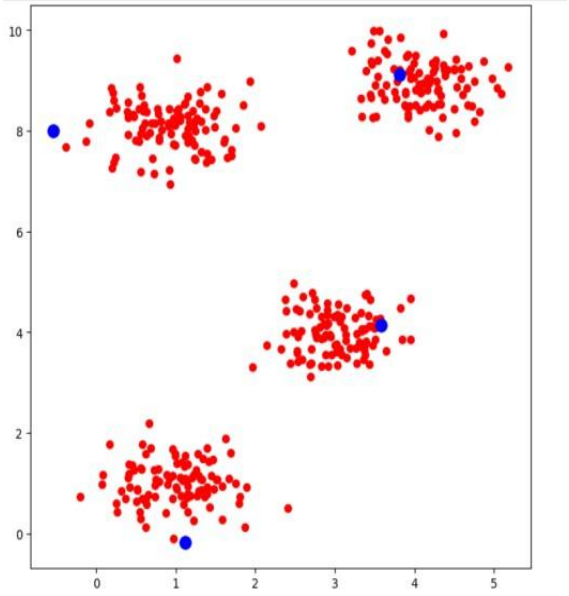
MODIFIED CODE:

Choosing first centroid randomly, choose second farthest away from first, third farthest away from first and second, and so on.

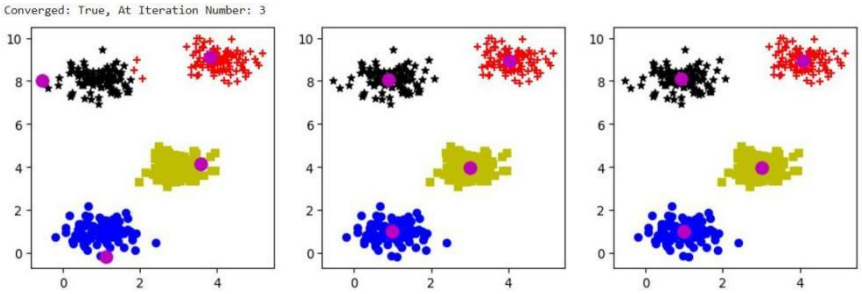
Applying a method to store the initial choice of centroids to ensure that the experiments are reproducible.

As we can see below the initial centroid are chosen randomly, and then we calculated the distance to select the second, third and fourth centroids making sure that the centroid are evenly distributed across the clusters which is better compared to our original code(clusters). I'm using pickle file to store the initial centroids to ensure the reproducibility and then the saved centroids are loaded back into the loaded_centroids.

```
Centroids have been saved to 'init_centroids.pkl'
Centroids are:
 [[ 3.81447065  9.11797161]
 [ 1.11546096 -0.1777632 ]
 [ 3.57869684  4.14133352]
 [-0.5399379   7.99961181]]
```



When we apply the above strategy, the algorithm converges in just 3 iterations which reduces the number of iterations required compared to our original code (random initialization of centroids).



Next, repeating the experiment with the modifications mentioned above for 10 times also for 4 centroids and calculating the number of iterations needed for the algorithm to converge for the same dataset.

By observing the below output the centroid initialization using the above strategy ensures that at each run the convergence is faster and consistent across all the iterations in each run. Finally, this strategy improves the efficiency of algorithm.

```
Run 1 with seed 22:
Converged: True, At Iteration Number: 3
Run 2 with seed 32:
Converged: True, At Iteration Number: 3
Run 3 with seed 42:
Converged: True, At Iteration Number: 3
Run 4 with seed 19:
Converged: True, At Iteration Number: 3
Run 5 with seed 18:
Converged: True, At Iteration Number: 3
Run 6 with seed 176:
Converged: True, At Iteration Number: 3
Run 7 with seed 24:
Converged: True, At Iteration Number: 3
Run 8 with seed 84:
Converged: True, At Iteration Number: 3
Run 9 with seed 39:
Converged: True, At Iteration Number: 2
Run 10 with seed 47:
Converged: True, At Iteration Number: 3
```

Conclusion:

The number of iterations to converge reduces on average across 10 runs when we use the modified centroid initialization method. This is because the modified code selects centroids using the farthest-point strategy, which reduces the risk of cluster overlap and ensures that the centroids are positioned and well-distributed properly across the dataset. This strategy leads to faster convergence compared to our traditional method, ensuring the better cluster assignments from the beginning.