

K-means clustering

import library

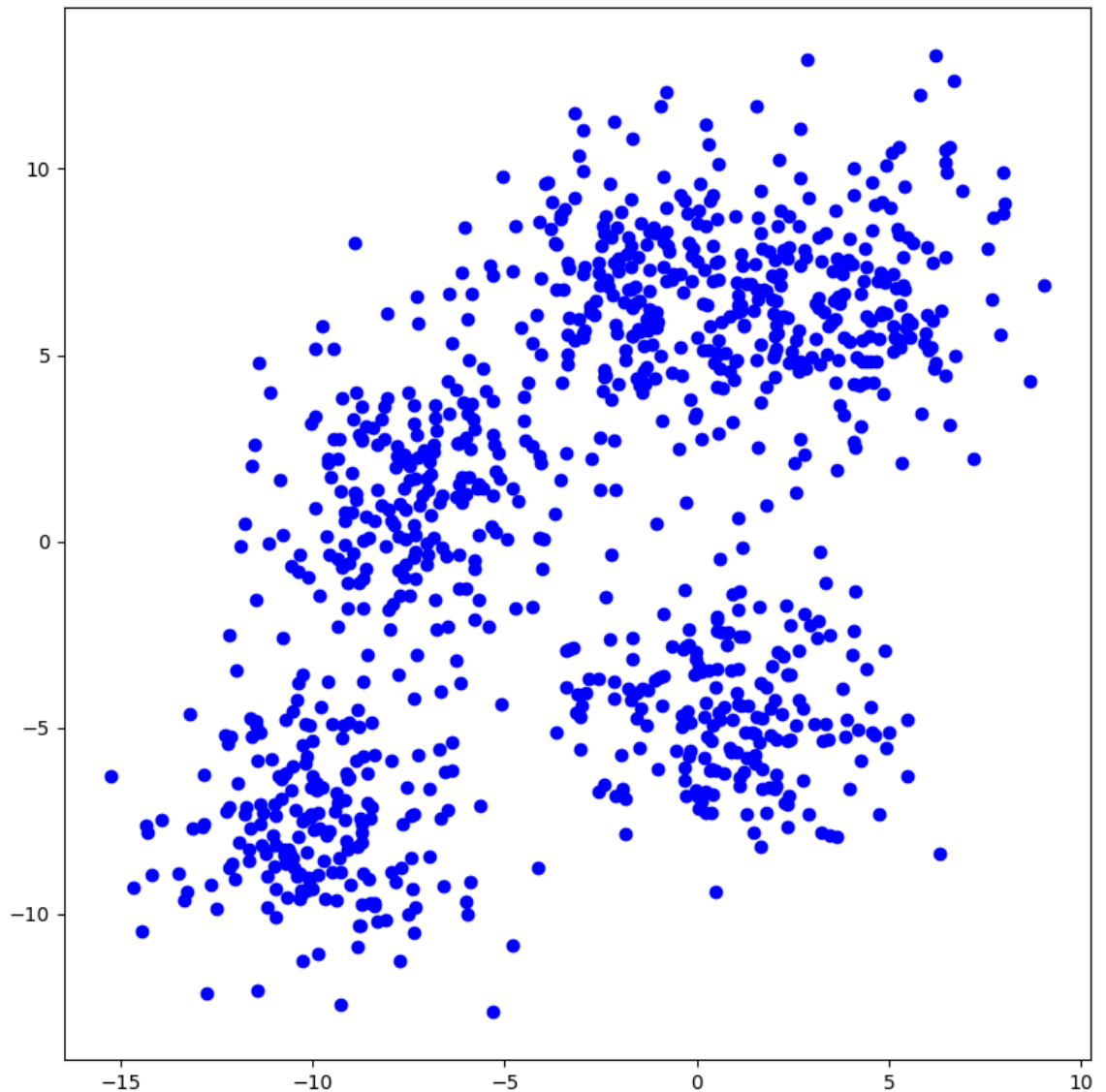
```
In [ ]: %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as colors
from matplotlib import cm
import time
import util
```

load data

```
In [ ]: fname_data = '11_data.csv'
data = np.genfromtxt(fname_data, delimiter=',')
num_data = np.size(data, 0)
num_feature = np.size(data, 1)
data = np.reshape(data, (num_data, num_feature))
```

plot the data

```
In [ ]: plt.figure(figsize=(8,8))
plt.scatter(data[:,0], data[:,1], color='blue')
plt.tight_layout()
plt.show()
```



optimization iteration

```
In [ ]: num_cluster    = 4
num_iter    = 50
```

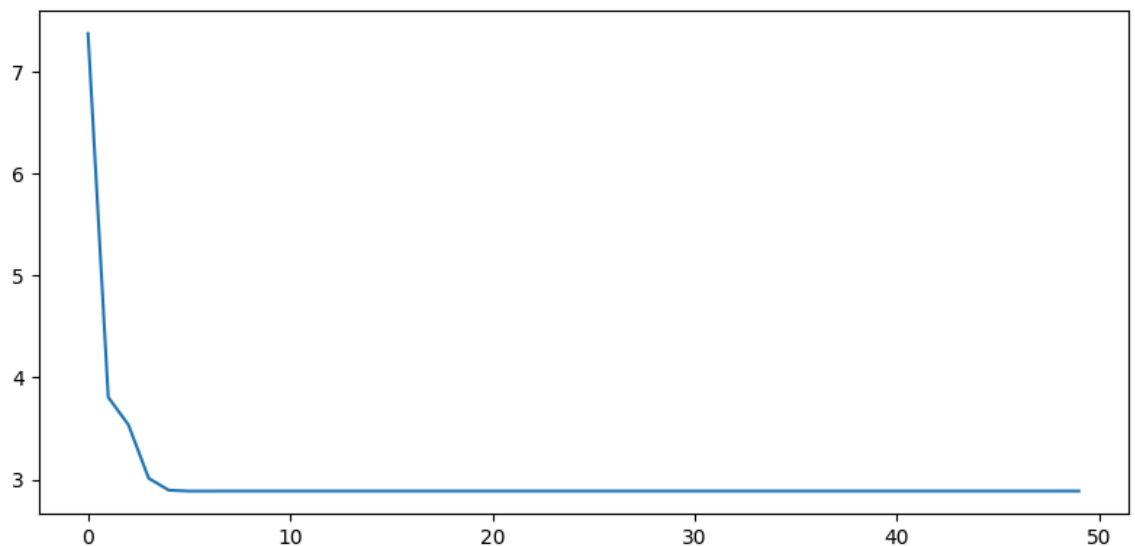
```
In [ ]: centroid       = np.zeros((num_cluster, num_feature))
centroid_iter  = np.zeros((num_iter, num_cluster, num_feature))
loss_iter      = np.zeros(num_iter)
label_data     = np.random.randint(0, num_cluster, size=(num_data)) # 0, 1, ..., num_cluster-1
distance       = np.zeros((num_data, num_cluster))
```

```
In [ ]: centroid_prev = util.compute_centroid(data, label_data, num_cluster, centroid)
```

```
In [ ]: for i in range(num_iter):
        centroid = util.compute_centroid(data, label_data, num_cluster, centroid_prev)
        distance = util.compute_distance(data, centroid)
        label_data = util.compute_label_data(distance)
        loss = util.compute_loss(distance, label_data)
        loss_iter[i] = loss
        centroid_iter[i] = centroid
        centroid_prev = centroid
```

```
In [ ]: def plot_01():
        plt.figure(figsize=(8,4))
        plt.plot(loss_iter)
        plt.tight_layout()
        plt.show()
        print('loss sum =', sum(loss_iter))
```

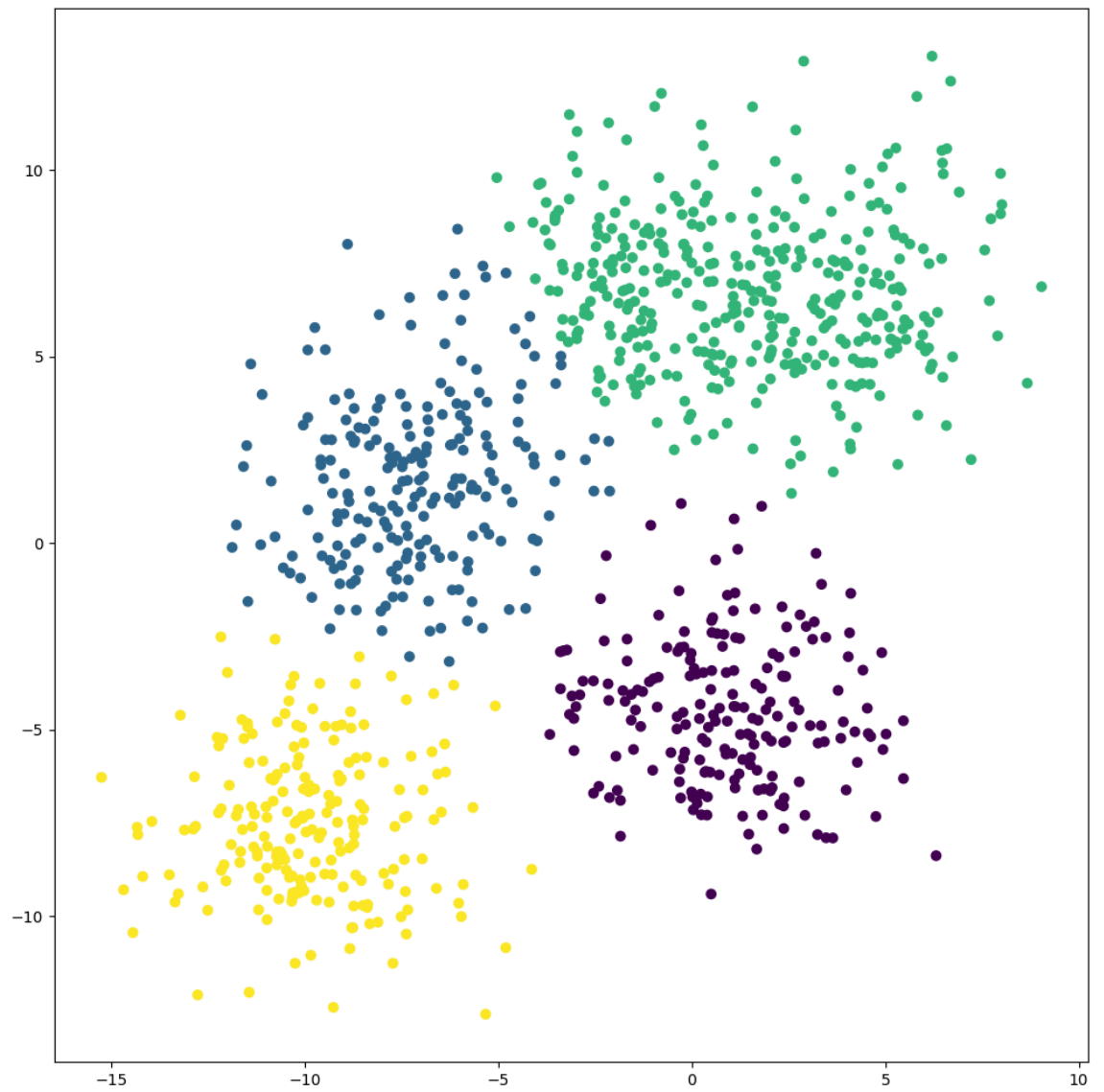
```
In [ ]: plot_01()
```



loss sum = 150.52526008611503

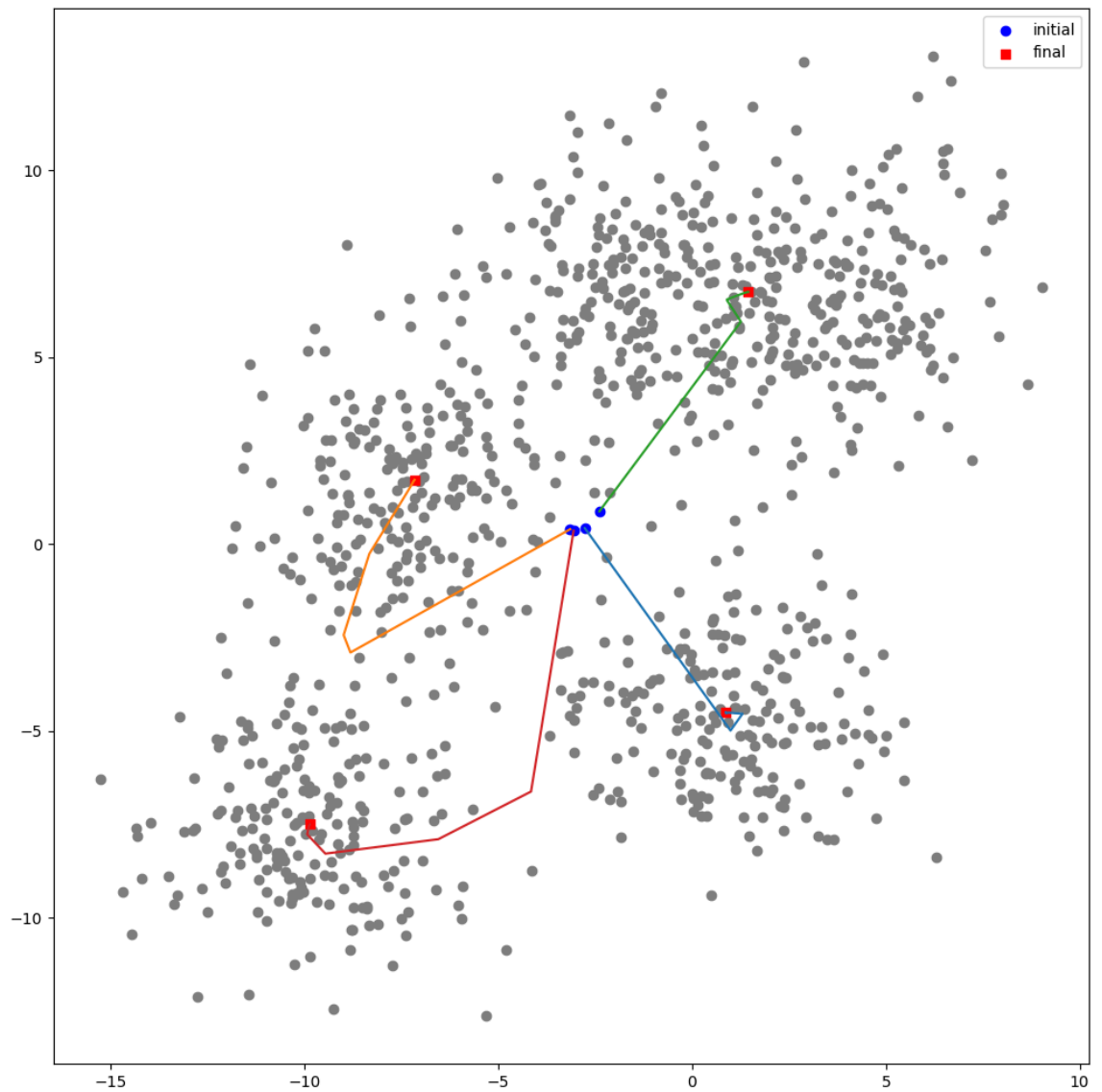
```
In [ ]: def plot_02():
        util.plot_data_label(data, label_data, num_cluster)
```

```
In [ ]: plot_02()
```



```
In [ ]: def plot_03():  
        util.plot_centroid(data, centroid_iter)
```

```
In [ ]: plot_03()
```



```
In [ ]: def plot_04():  
        for i in range(num_cluster):  
            print('label :', i, 'centroid :', centroid[i,:])
```

```
In [ ]: plot_04()
```

```
label : 0 centroid : [ 0.86873053 -4.5026988 ]  
label : 1 centroid : [-7.15750935  1.71741107]  
label : 2 centroid : [1.45197246  6.74154716]  
label : 3 centroid : [-9.86048003 -7.49938861]
```

results

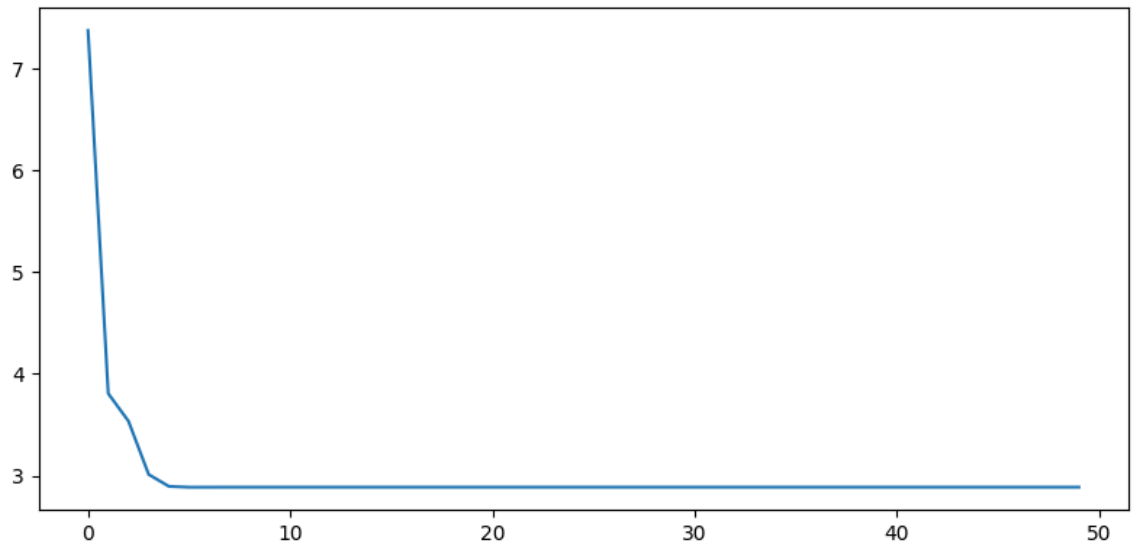

```
In [ ]: number_result = 4

for i in range(number_result):
    title = '# RESULT # {:02d}'.format(i+1)
    name_function = 'plot_{:02d}()'.format(i+1)

    print('')
    print('#####')
    print(title)
    print('#####')
    print('')

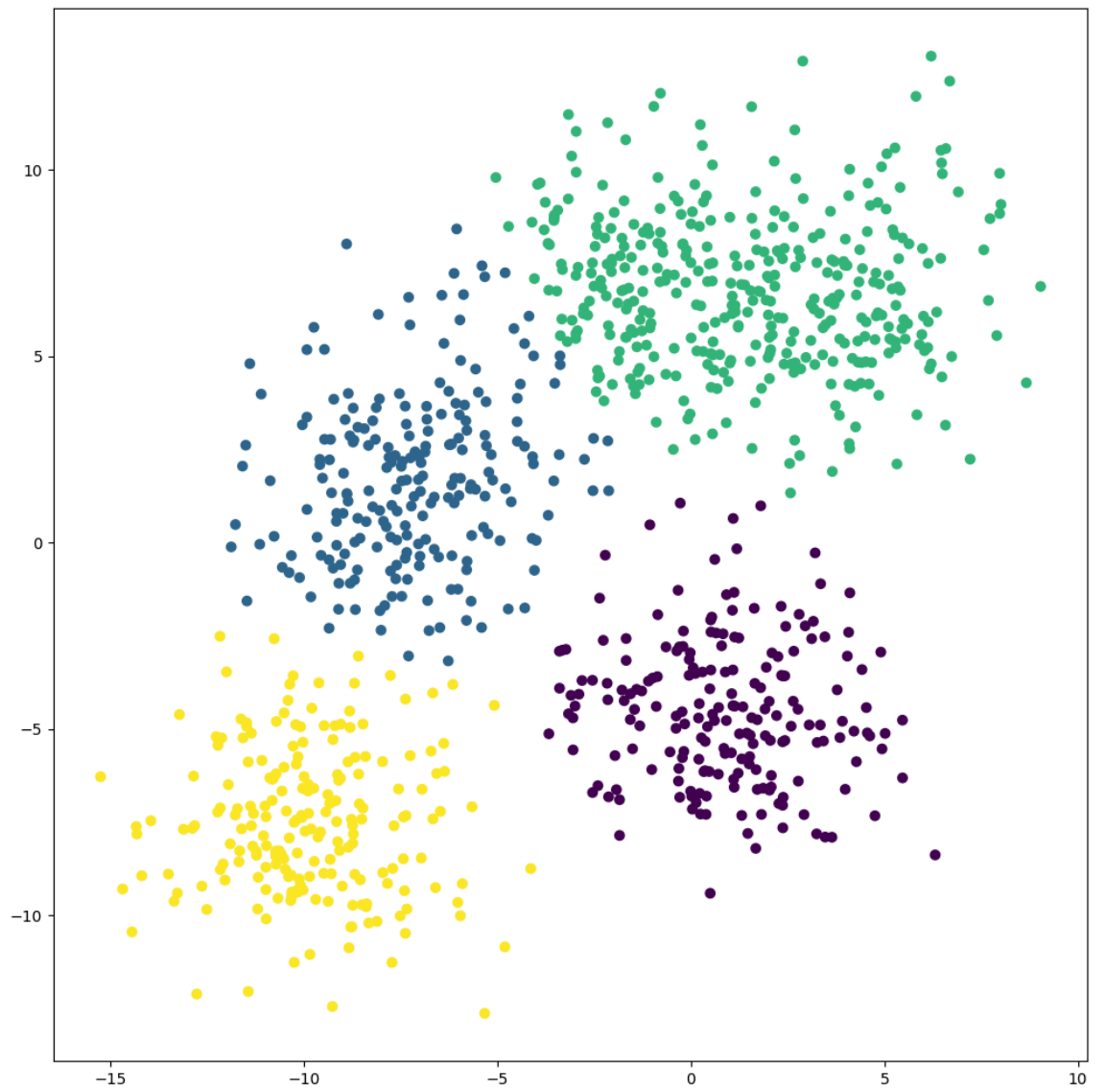
    eval(name_function)
```

```
#####  
#####  
# RESULT # 01  
#####  
#####
```

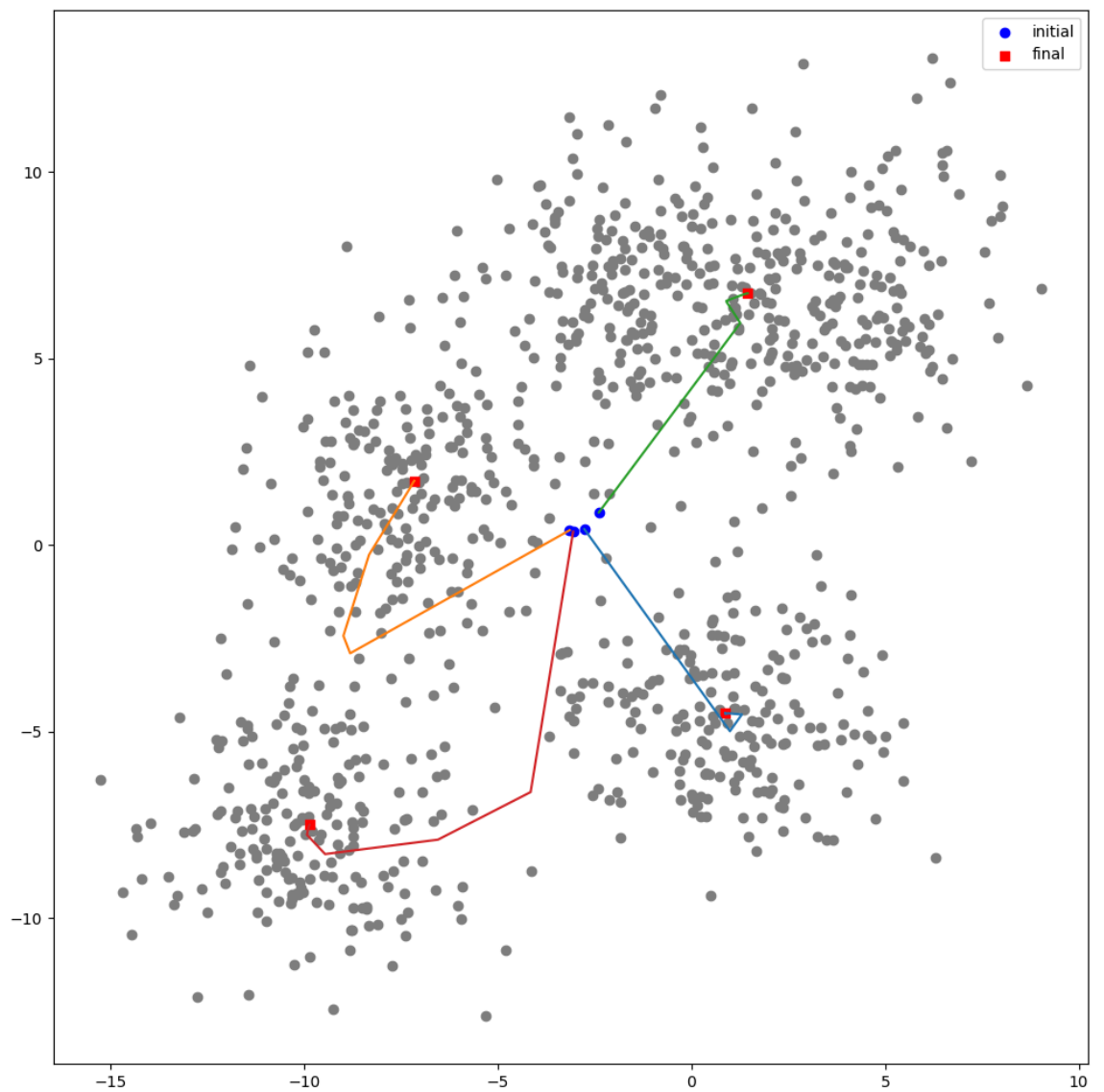


loss sum = 150.52526008611503

```
#####  
#####  
# RESULT # 02  
#####  
#####
```

```
#####  
#####  
# RESULT # 03  
#####  
#####
```



```
#####  
#####  
# RESULT # 04  
#####  
#####
```

```
label : 0 centroid : [ 0.86873053 -4.5026988 ]  
label : 1 centroid : [-7.15750935  1.71741107]  
label : 2 centroid : [1.45197246  6.74154716]  
label : 3 centroid : [-9.86048003 -7.49938861]
```