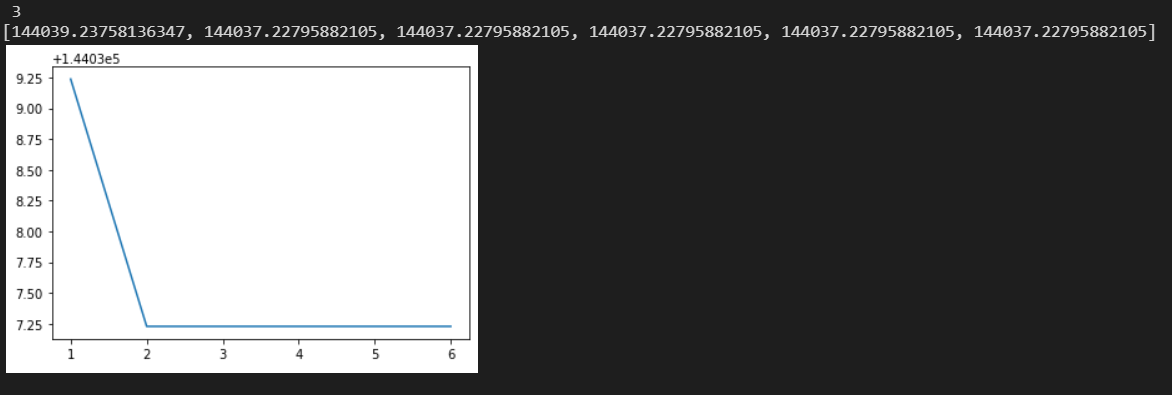
2. a. (i)

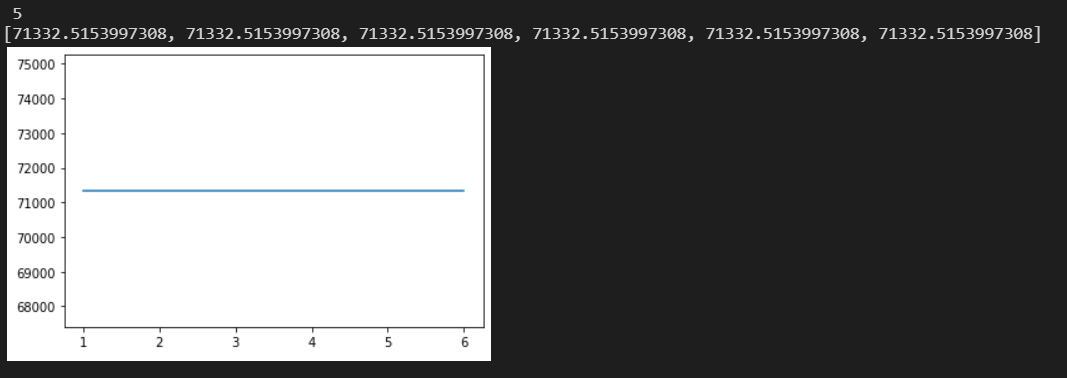
For k = 3

Total SSE of each clustering run



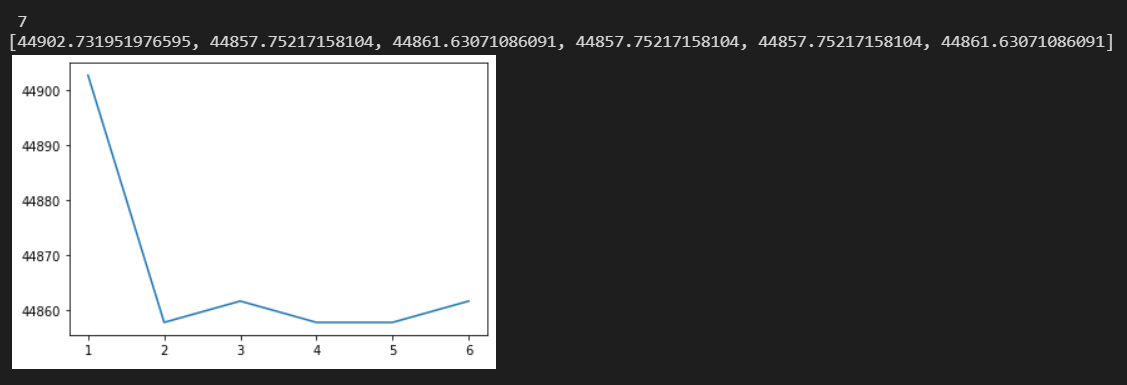
For k = 5

Total SSE of each clustering run



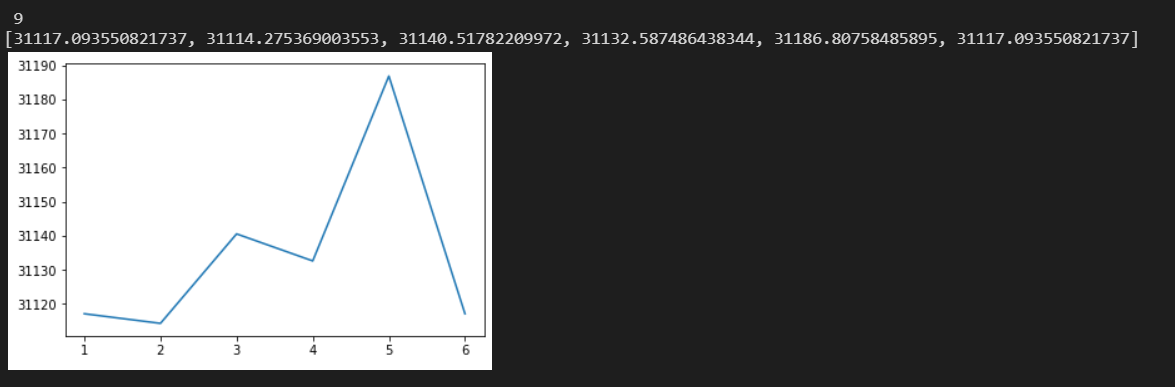
For k = 7

Total SSE of each clustering run



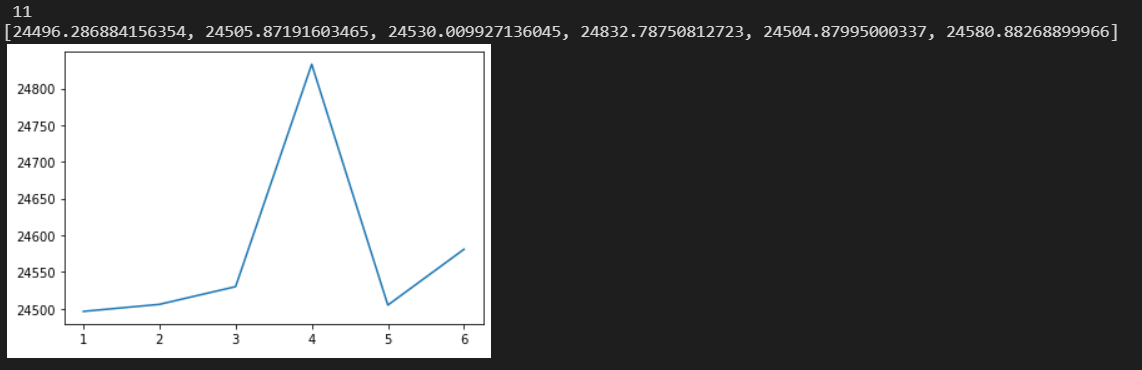
For k = 9

Total SSE of each clustering run

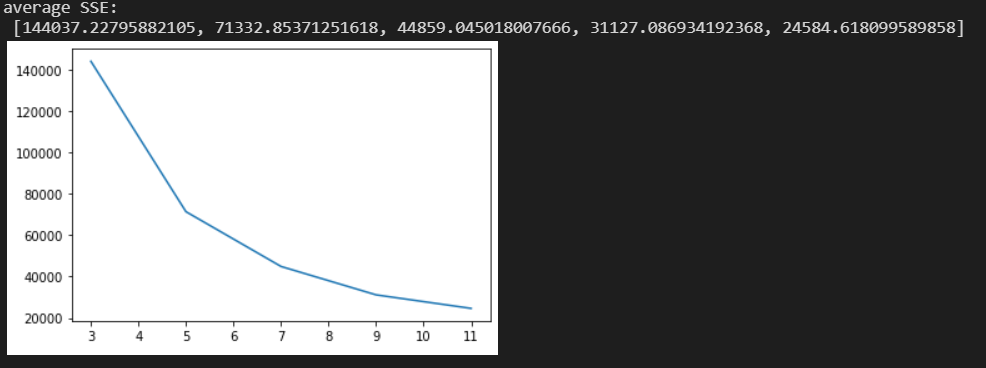


For k = 11

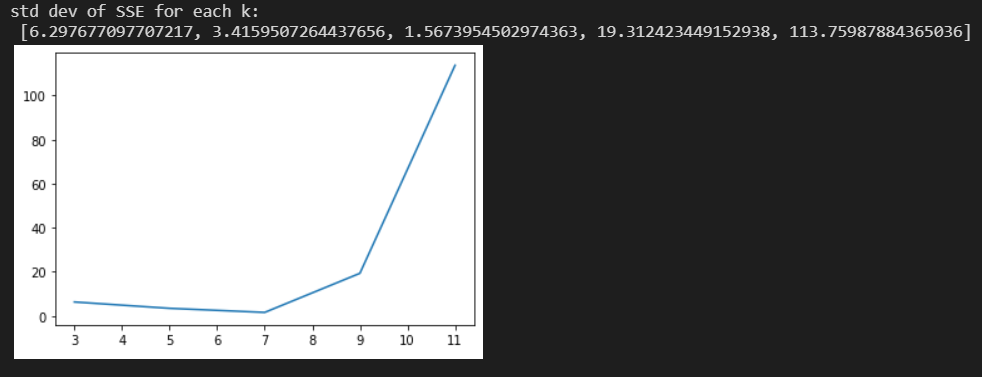
Total SSE of each clustering run



(ii)Average SSE for each k

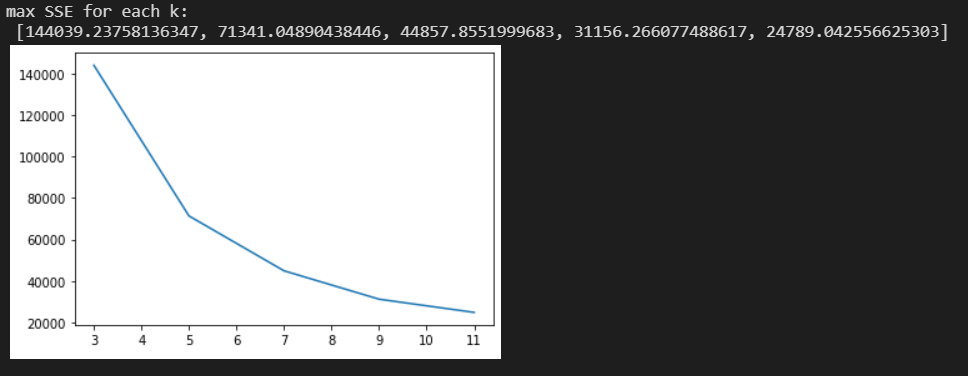


Standard deviation of SSE for each k

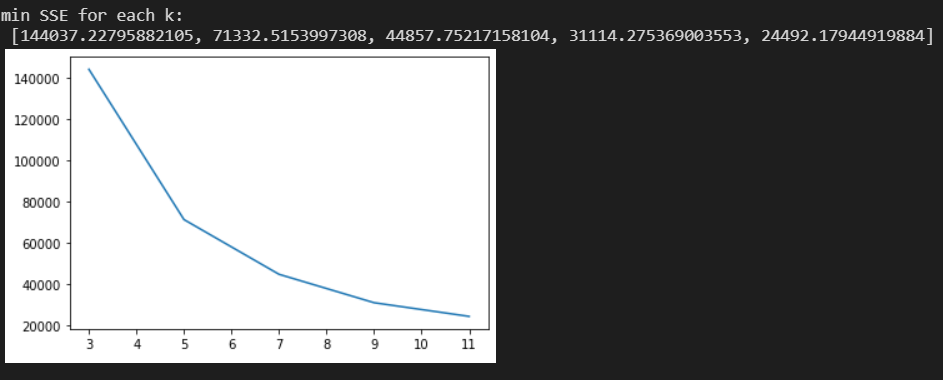


(iii) Min and max SSE for each k

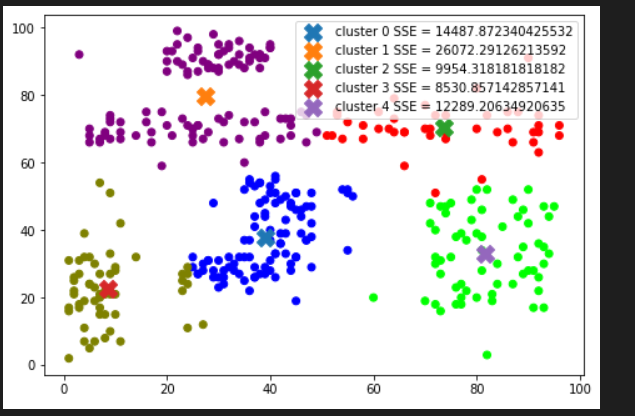
Max SSE for each k



Min SSE for each k



b.



c.

The top cluster should be separated with the belt cluster, and the belt region in the middle should be in a single cluster.

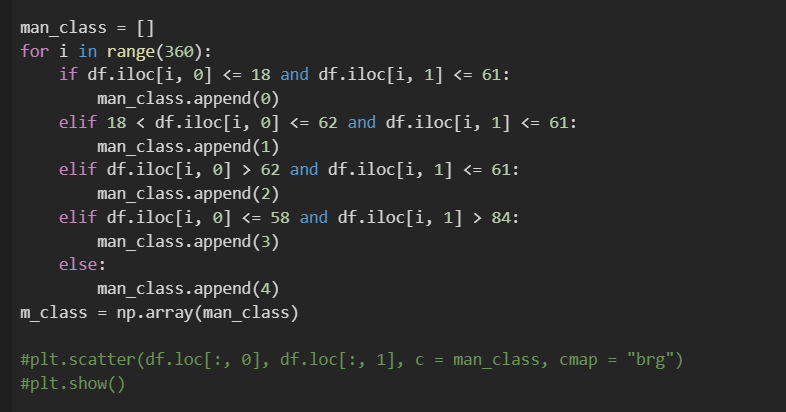
Because the Euclidean distance is used, kmeans cannot correctly cluster the belt region in the middle. This proved that the k means clustering doesn’t work well if the data is not spherically distributed.

d.

These data can be classified with linear boundaries, since the margin between each cluster is rathe large.

A screenshot of a cell phone

Description automatically generated



e.

Construct a contingency matrix, which is symmetric

k\_means = cluster

manual clustering = class

nij = intersection(class i, cluster j)

nij = nji

class/cluster 0 1 2 3 4

0 n00 n01 n02 n03 n04

1 n11 n12 n13 n14

2 n22 n23 n24

3 n33 n34

4 n44

rand\_idx = (a + d) / (a + b + c + d)

a = sigma(comb(nij, 2)

b = sigma(comb(ni., 2)) - sigma(comb(nij, 2))

c = sigma(comb(n.j, 2)) - sigma(comb(nij, 2))

d = comb(N, 2) - a - b – c

N is the total number of data points

**rand index = (a + d) / (a + b + c + d) = 0.8963478799133395**

The rand index represents the accuracy of the target clustering, which is kmeans in this case. (Or the similarity between two clustering.)

