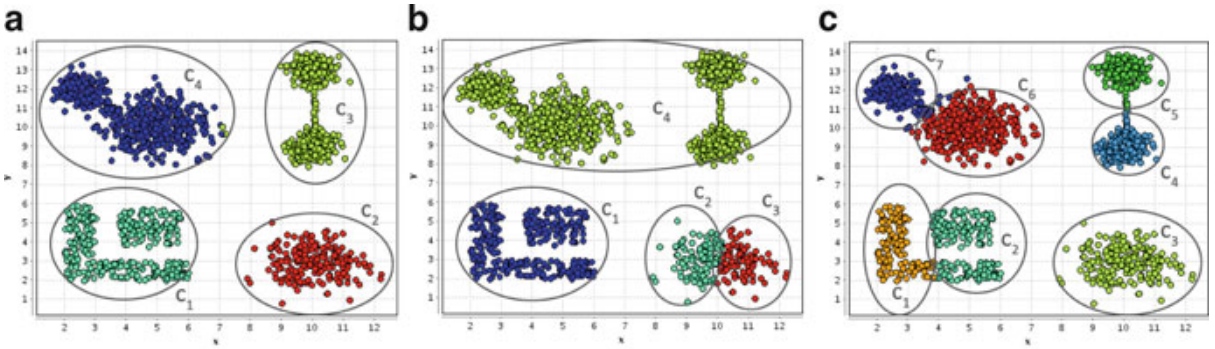
K means clustering

Algorithm:

- Select K random data points as the centers of K clusters
- Assign each datapoint to the closest clusters (by calculating the distance from
- centers).
- While **True**:
  - Recalculate the center of the clusters (which is the mean of the data points)
  - Reassign each datapoint to the closest cluster
  - **If** no datapoint changes cluster **then**
    - **break**

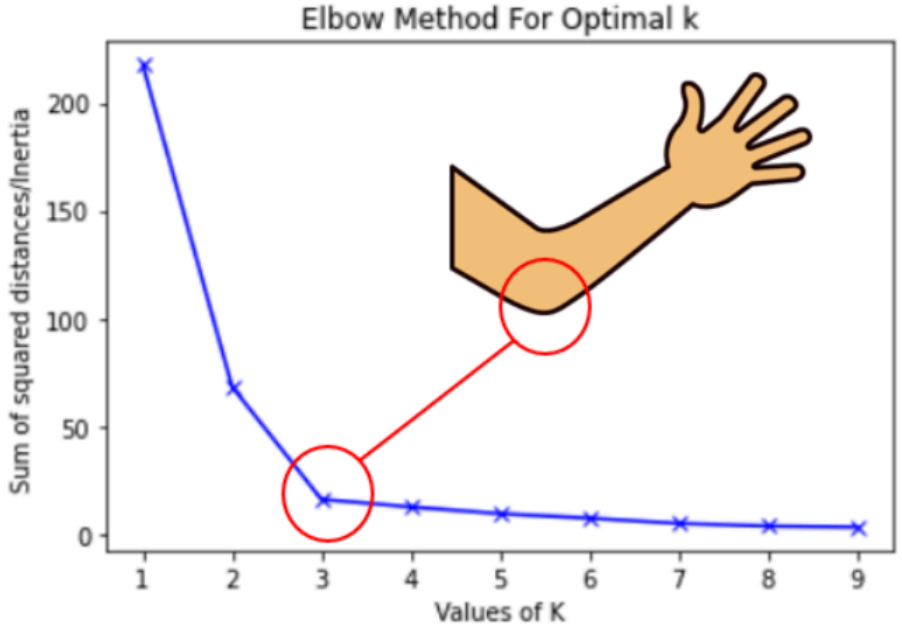
Limitations:

- Need to know K in advance
- Depended on initial assignment of the centers



Elbow method:

Inertia measures how well a dataset was clustered by K-Means. It is calculated by measuring the distance between each data point and its centroid, squaring this distance, and summing these squares across one cluster. A good model is one with low inertia AND a low number of clusters (K).



Line plot between K and inertia

Problem 1:

- Apply K-means clustering algorithm on the dataset *'dataset.txt'* in the given folder.
- Plot the data for different values of K, note down the inertia and apply *Elbow method* to find the suitable value for K.
- You must follow the given algorithm.
- Do not use libraries such as : *sklearn*, *scikit learning* or *pandas* for this assignment. You need to implement from scratch. Refer to the python notebook given on eLMS if you need help.
- Submit a *.ipynb* file.

Problem 2:

- Apply K-means clustering algorithm on the color space of an image to compress it.
- Use K = 8, 16, 32 and plot these images to compare different compression levels.
- You can use any image of your choice for this problem.
- You can use libraries for this problem.
- Submit a *.ipynb* file.
- You will need to demonstrate your code. Your marks will depend on your viva and understanding.