**Applying K-means clustering to divide the data**

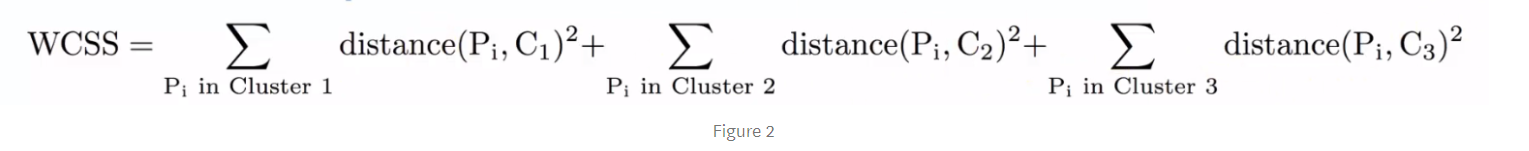
* **How to find ‘K’?**

The value of k determines how many clusters there are going to be in the resulting cluster distribution. We can choose any number of clusters as we want.

But there is a standard procedure to find the optimal number of clusters. This method using **W**ithin **C**lusters **S**um of **S**quares (**WCSS)**.

In this method, we take various numbers of k (I have chosen from 1 – 20), and then find the value of WCSS for each number of k. Then we plot the WCSS wrt number of k.

Afterwards we observe the graph and take the number of K on which the graph seems to ‘level-out’ afterwards. This point is known as elbow point.

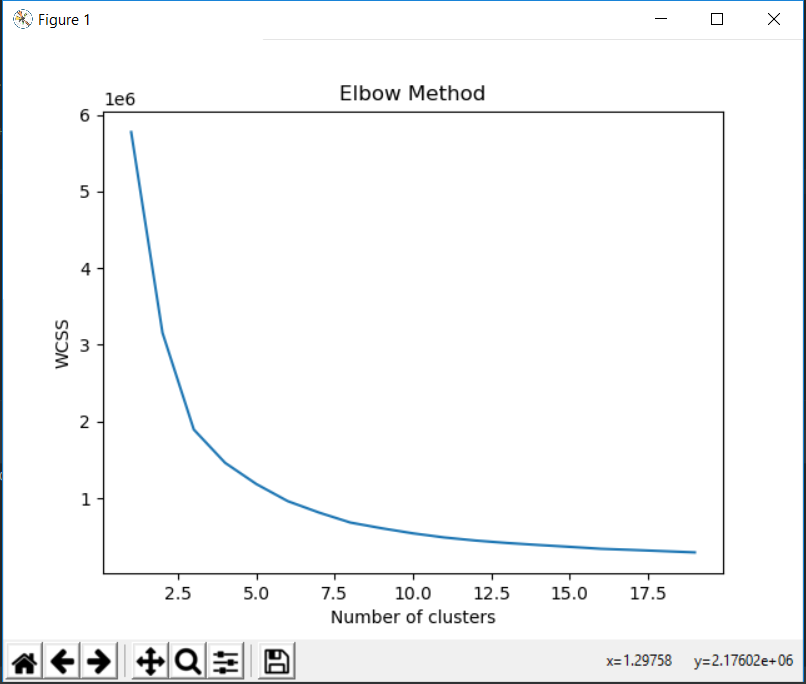


I have used K = 1 to 20 for the trial and the code for this purpose is given as:

**find\_k.py**

**(Note: Make sure to keep the “final\_points.txt” in the same folder as of find\_k.py)**

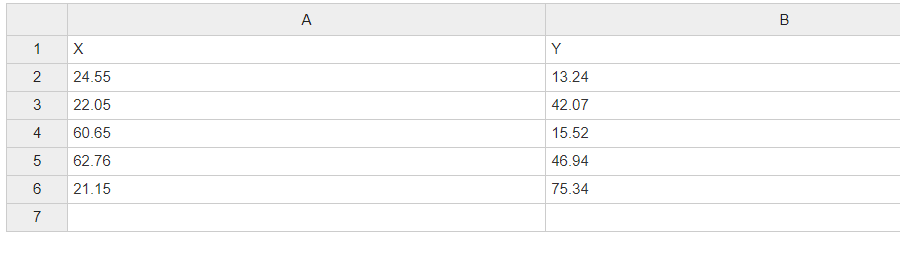
After running this given code, the plot is as under:



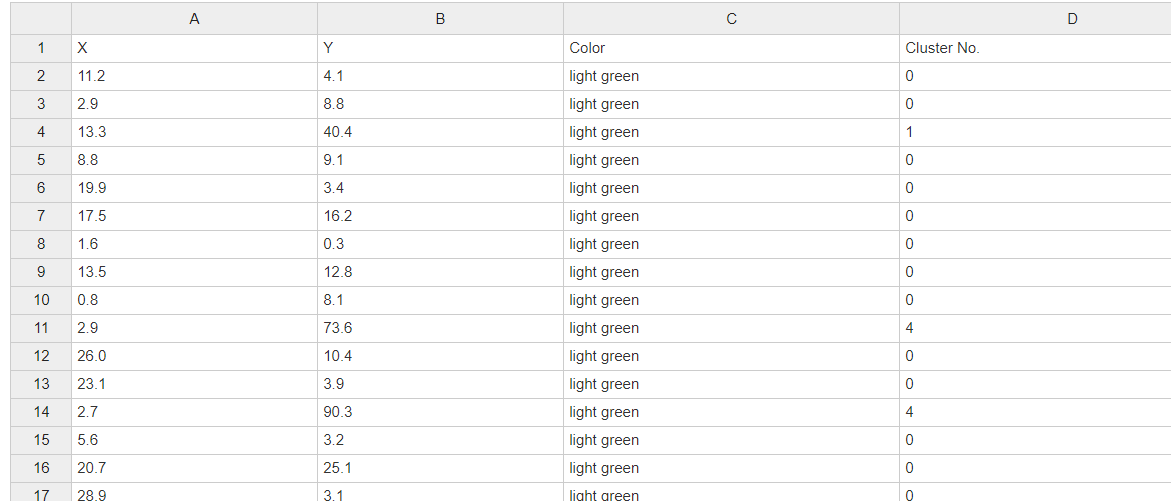
Here we can see that graph is levelling out after 5. So lets take K = 5.

* **Results:**

The locations of 5 clusters’ centers can be found in “**centers.txt**”. Here is a glimpse:



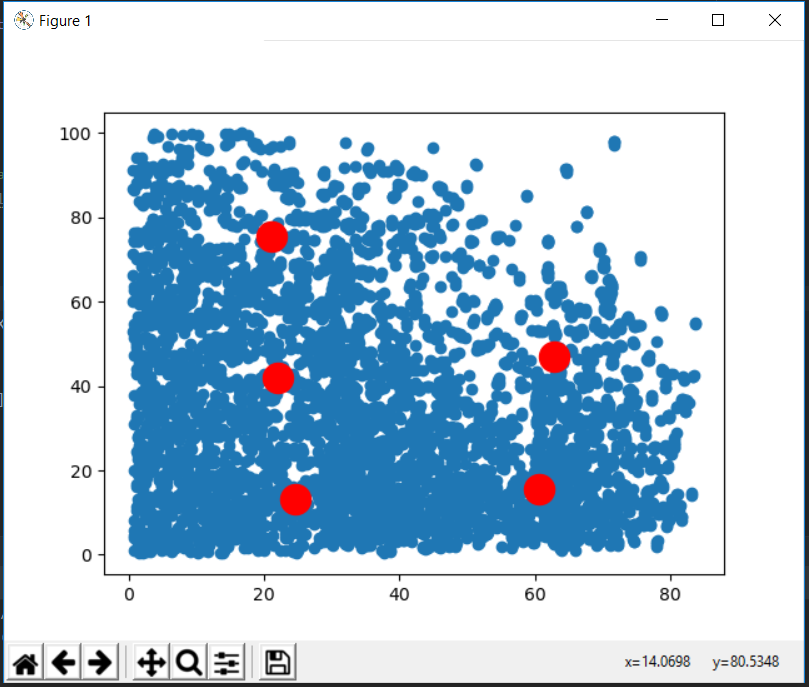
I have also applied this model to make predictions on all of the data. All the classified data can be found in **“clusterred\_data.txt”.**



**(Cluster numbers are 0-4)**

The code for K-means clustering is given as: **“kmeans\_implementation.py”**

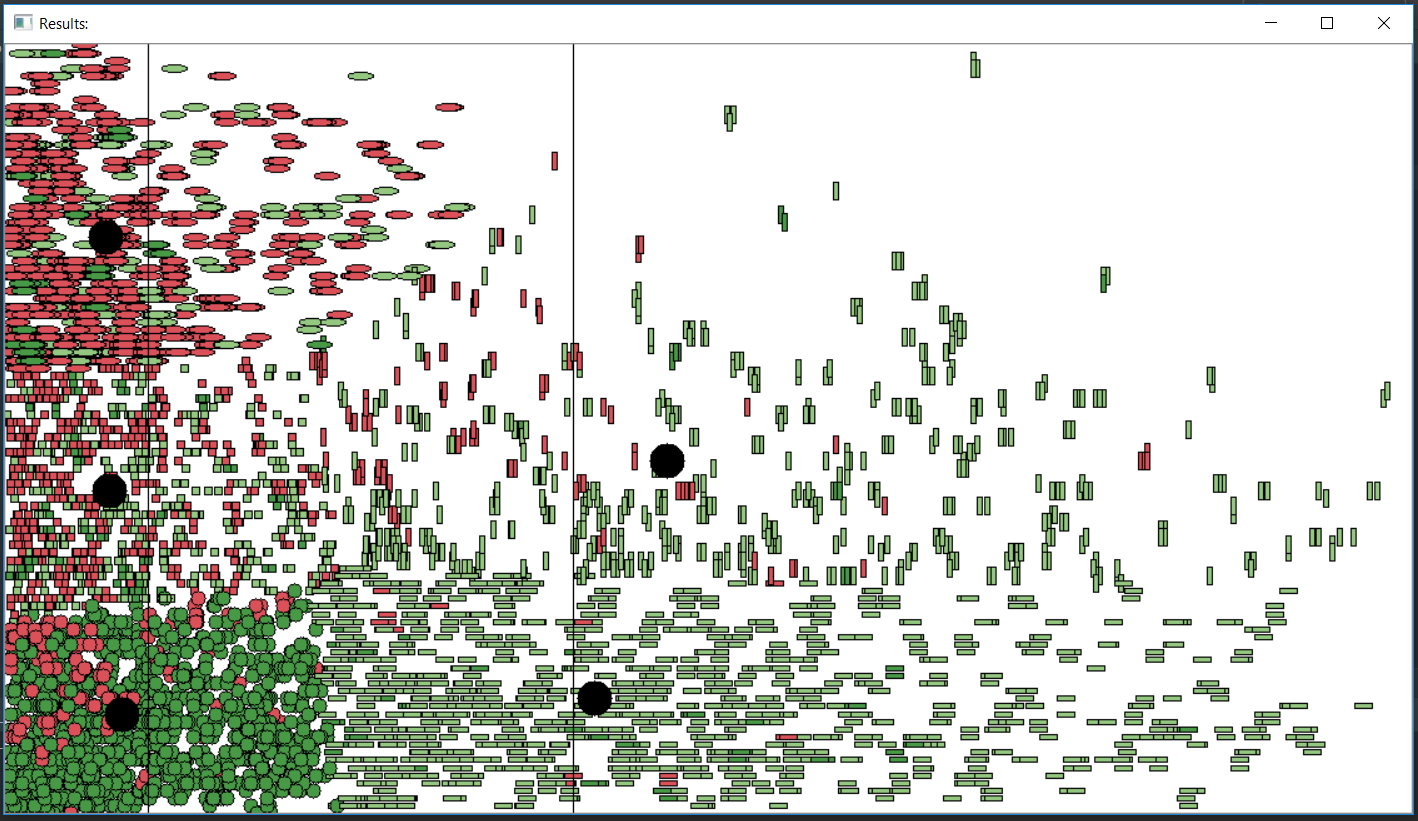
The resulting scatter plot is:



(This monochromatic plot shows all data points of 1 clr)

For better visualization, please use the code “**plotting\_results.py”** (after running the kmeans\_implementation.py code as it reads the files created by kmeans\_implementation.py)

Visualization:



(I have also sent the HD images on upwork)

Here,

* Black circles are the centeroids of 5 clusters.
* Cluster 0 is shown by small circles.
* Cluster 1 is shown by small squares.
* Cluster 2 is shown by LONG rectangles.
* Cluster 3 is shown by HIGH rectangles.
* Cluster 4 is shown by Ellipsoids.

So that’s it 😊