Question 1: Assignment Summary

Problem Statement:

- HELP International is an international humanitarian NGO that is committed to fighting poverty and providing the people of backward countries with basic amenities and relief during the time of disasters and natural calamities. It runs a lot of operational projects from time to time along with advocacy drives to raise awareness as well as for funding purposes.
- After the recent funding programs, they have been able to raise around $ 10 million. Now the CEO of the NGO needs to decide how to use this money strategically and effectively. The significant issues that come while making this decision are mostly related to choosing the countries that are in the direst need of aid.

Objective:

- Objective is to segment the countries using some socio-economic and health factors that determine the overall development of the country.

- Then we need to suggest the countries which the CEO needs to focus on the most for considering for NGO Aid.

Method followed:

- **Data Understanding and Processing:**

  - ✓ Checked the total number of rows and columns in data frame
  - ✓ Checked datatypes of each columns and no need to perform datatype conversion as all columns were having correct datatype
  - ✓ It was found that there were no null values
  - ✓ There were also no duplicate values for country
  - ✓ There were a few outliers found and they were treated later before clustering
  - ✓ Performed soft capping (1% to 99%) for few features which are having very few extreme outliers
  - ✓ We have performed scaling on all numerical data columns, so that all features will get equal weightage during clustering

- **Data visualization (EDA):**

  - ✓ We have performed Univariate analysis like Distribution plot to check data distribution, Box plot for outlier analysis
  - ✓ We have performed Bivariate analysis like Bar plot between country and different numerical features to get some understanding of which country requires more NGO AID

- **Clustering:**

  - ✓ Checked Hopkin statistics multiple times and got Hopkins value close to 1 each time. As we know, if Hopkins score closes to 1, there is a good cluster tendency
  - ✓ Plot Elbow curve and Silhouette score and got the optimal K value as 3
  - ✓ Perform both K-Means clustering and Hierarchical clustering and able to segment country dataset among Under Developed, Developing and Developed country.
  - ✓ We got almost similar top 10 Under Developed country names, which need NGO AID, from both K-Means and hierarchical clustering.
  - ✓ We have observed Hierarchical Clustering is more prone to Outliers. Presence of any extreme outliers, which is not capped has effect on cluster formation.
  - ✓ The Result of K-Means is more stable to be considered for final NGO AID.

Question 2: Clustering

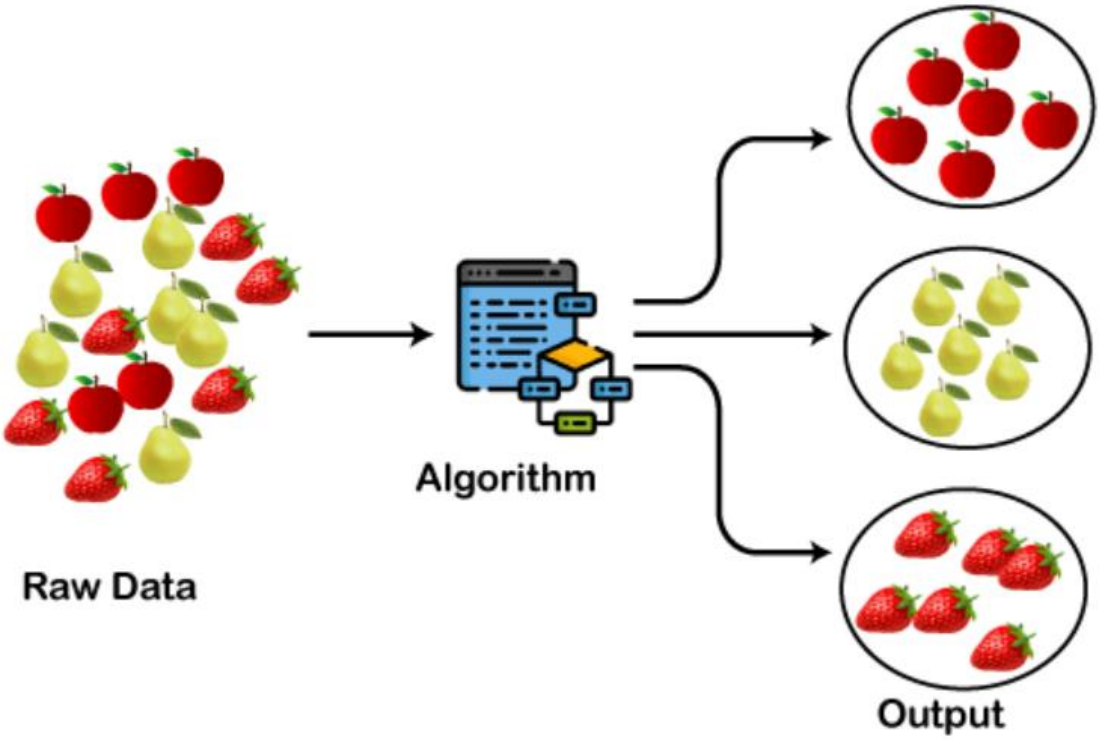
a) Compare and contrast K-means Clustering and Hierarchical Clustering.

Clustering is a machine learning technique, which groups the unlabeled dataset. It can be defined as ***"A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group."***

It performs segmentation by finding some similar patterns in the unlabeled dataset such as shape, size, color, behavior, etc., and divides them as per the presence and absence of those similar patterns.

It falls under **Unsupervised machine learning method**, as it deals with the unlabeled dataset and there is no target variable.

The below diagram explains the working of the clustering algorithm. We can see the different fruits are divided into several groups with similar properties.



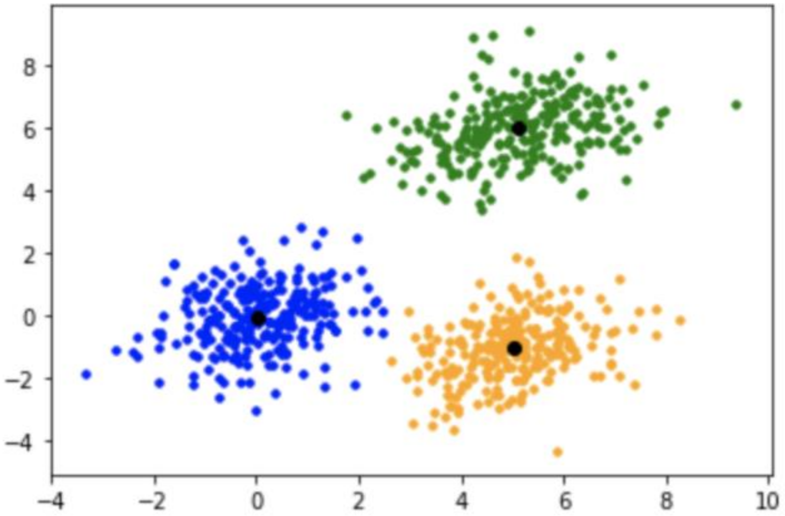
Among the **different types** of available clustering methods, we will discuss about below two methods:

- **K-means Clustering**
- **Hierarchical Clustering**

K-means Clustering: It is the simplest unsupervised learning algorithm that solves clustering problem. K-means algorithm partition n observations into k clusters where each observation belongs to the cluster with the nearest mean serving as a prototype of the cluster. K-means is a centroid-based algorithm, where each cluster is associated with a centroid and we calculate the distances to assign a point to a cluster.

The main objective of the K-Means algorithm is to minimize the sum of squared distances between the points and their respective cluster centroid.

Below clusters formed by a K-means clustering algorithm.

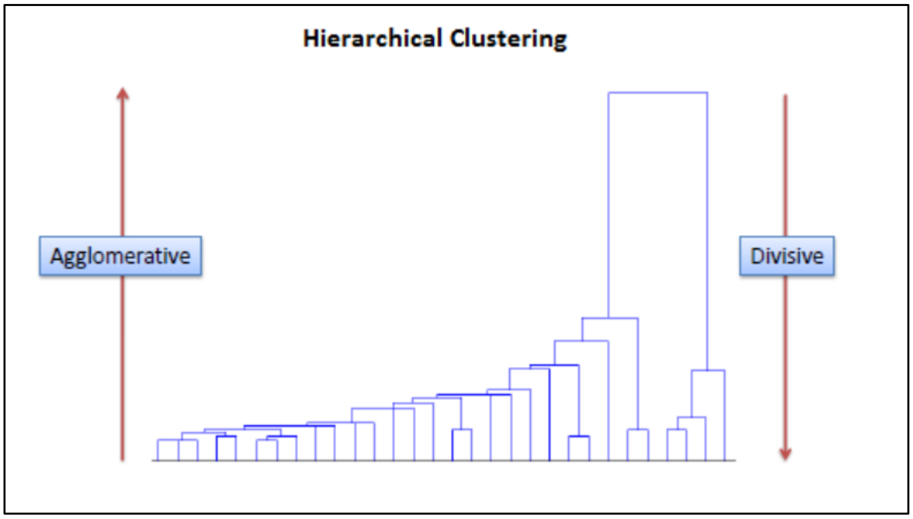


Hierarchical Clustering: Hierarchical clustering means creating a tree of clusters by iteratively grouping or separating data points. One of the advantages of hierarchical clustering is that we do not have to specify the number of clusters.

There are **<u>two types</u>** of hierarchical clustering:

1. **Agglomerative clustering:** Agglomerative clustering is kind of a **<u>bottom-up approach</u>**. Each data point is assumed to be a separate cluster at first. Then the similar clusters are iteratively combined and finally make one big cluster containing all data pints.

2. **Divisive clustering:** Divisive clustering is the opposite of agglomerative clustering. We start with one giant cluster including all data points. Then data points are separated into different clusters. It is a **<u>top to bottom</u>** approach.



k-means Clustering	Hierarchical Clustering
k-means, using a pre-specified number of clusters, the method assigns records to each cluster to find the mutually exclusive cluster of spherical shape based on distance. i.e., K Means clustering needed advance knowledge of K.	In hierarchical clustering one can stop at any number of clusters, one finds appropriate by interpreting the dendrogram. Hierarchical methods can be either divisive or agglomerative.
One can use median or mean as a cluster center to represent each cluster.	Agglomerative methods begin with 'n' clusters and sequentially combine similar clusters until only one cluster is obtained.
Methods used are normally less computationally intensive and are suited with very large datasets.	Hierarchical methods work in the opposite direction, Hierarchical methods are especially useful when the target is to arrange the clusters into a natural hierarchy.
In K Means clustering, since one start with random choice of clusters, the results produced by running the algorithm many times may differ.	In Hierarchical Clustering, results are reproducible in Hierarchical clustering
K- means clustering a simply a division of the set of data objects into non- overlapping subsets (clusters) such that each data object is in exactly one subset).	A hierarchical clustering is a set of nested clusters that are arranged as a tree.
K Means uses sum of squared distance between data point and nearest centroid as the cost function.	There is no cost function associated with Hierarchical clustering. Business ill decide threshold for K.

Advantages of K-Means:

- **Simple:** It is easy to implement k-means and identify unknown groups of data from complex data sets.
- **Flexible:** K-means algorithm can easily adjust to the changes. If there are any problems, adjusting the cluster segment will allow changes to easily occur on the algorithm.
- **Suitable in a large dataset:** K-means is suitable for large number of datasets and it's computed much faster than the smaller dataset. It can also produce higher clusters.

- **Efficient:** The algorithm used is good at segmenting the large data set. Its efficiency depends on the shape of the clusters.
- **Tight clusters:** Compared to hierarchical algorithms, k-means produce tighter clusters especially with globular clusters.
- **Computation cost:** Compared to using other clustering methods, a k-means clustering technique is fast and efficient in terms of its computational cost.
- **Spherical clusters:** This mode of clustering works great when dealing with spherical clusters.
- **Guarantees convergence:** This guarantees convergence with in finite number of steps.

Disadvantages of K-Means:

- For K-means clustering to be effective, you must specify the number of clusters (K) at the beginning of the algorithm.
- Different initial partitions can result in different final clusters.
- K-Means is sensitive to outliers. Centroids can be dragged by outliers, or outliers might get their own cluster instead of being ignored. Consider removing or clipping outliers before clustering.
- Scaling with number of dimensions.
- As the number of dimensions increases, a distance-based similarity measure converges to a constant value between any given examples. Reduce dimensionality either by using PCA on the feature data, or by using “spectral clustering” to modify the clustering algorithm as explained below.
- It does not work well with clusters (in the original data) of Different size and Different density.

Advantages of Hierarchical Clustering:

- Hierarchical clustering outputs a hierarchy, i.e. a structure that is more informative than the unstructured set of flat clusters returned by k-means. Therefore, it is easier to decide on the number of clusters by looking at the dendrogram.
- Easy to implement

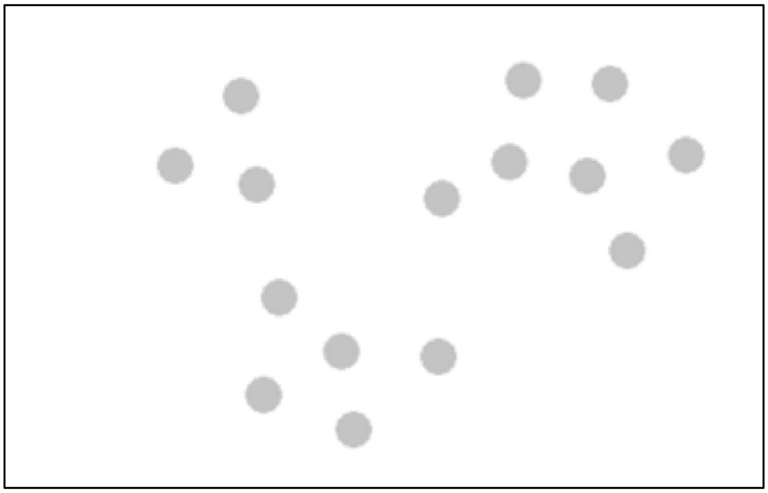
Disadvantages of Hierarchical Clustering:

- It is not possible to undo the previous step. Once the instances have been assigned to a cluster, they can no longer be moved around.
- Time and space complexity high: not suitable for large datasets.
- Very sensitive to outliers

b) Briefly explain the steps of the K-means clustering algorithm.

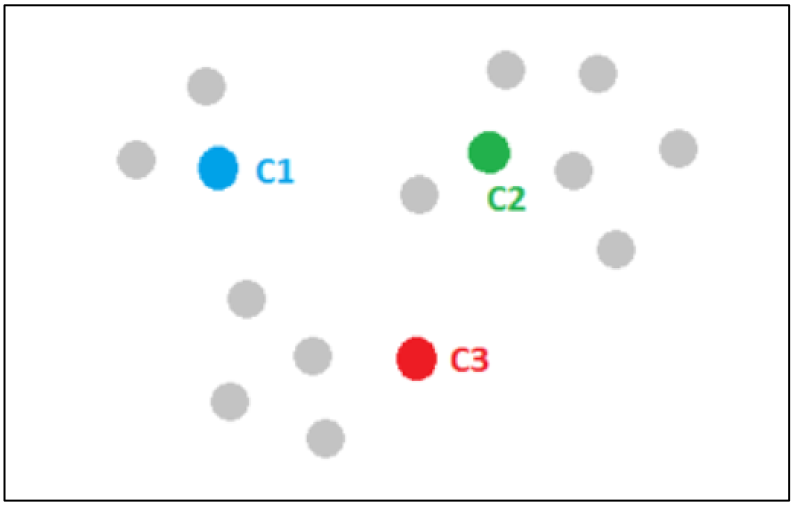
Among all unsupervised learning algorithms, clustering via k-means might be one of the simplest and most widely used algorithms. Briefly speaking, k-means clustering aims to **find the set of k clusters such that every data point is assigned to the closest center, and the sum of squared distances of all such assignments is minimized**.

Let’s take an example to better understand the idea. Imaging we have these 2D gray points in the following figure and want to assign them into three clusters. K-means follows the **four** steps listed below.

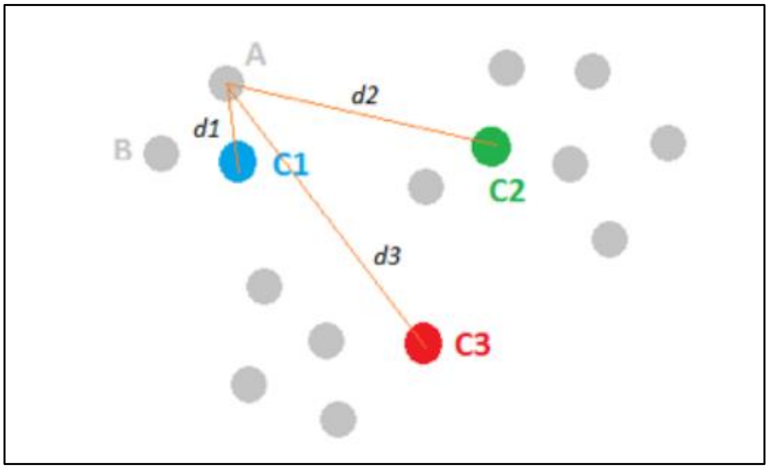


Step one: *Initialize cluster centers*

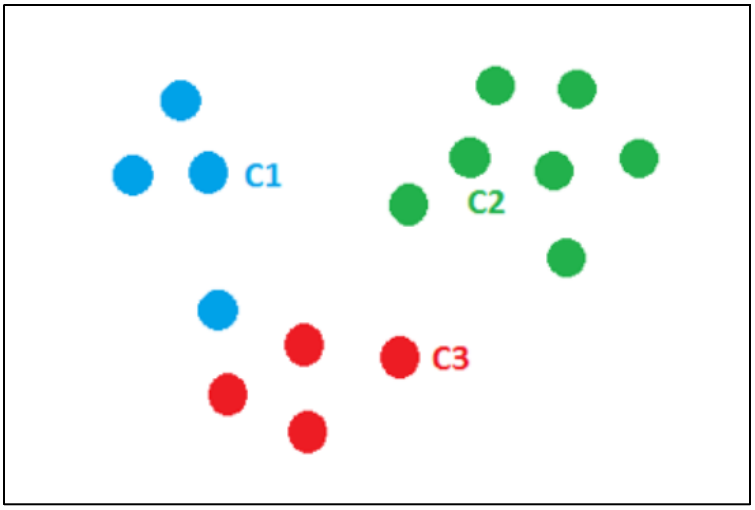
We randomly pick three points **C1**, **C2** and **C3**, and label them with blue, green and red color separately to represent the cluster centers. Here we have considered K as 3, that’s why chosen three cluster centers.



Step two: *Assign observations to the closest cluster center*



Once we have these cluster centers, we can assign each point to the clusters based on the **minimum distance to the cluster center**. For the gray point A, compute its distance to **C1**, **C2** and **C3**, respectively. And after comparing the lengths of ***d1***, ***d2*** and ***d3***, we figure out that *d1* is the smallest, therefore, we assign point A to the blue cluster and label it with blue. We then move to point B and follow the same procedure. This process can assign all the points and leads to the following figure.



This step is called ***Assignment step*** as we assign every data point to K clusters.

The **equation for the assignment step** is as follows:

$$\boxed{Z_i = argmin||X_i - \mu_k||^2}$$

Where, X_i is the i^{th} data point

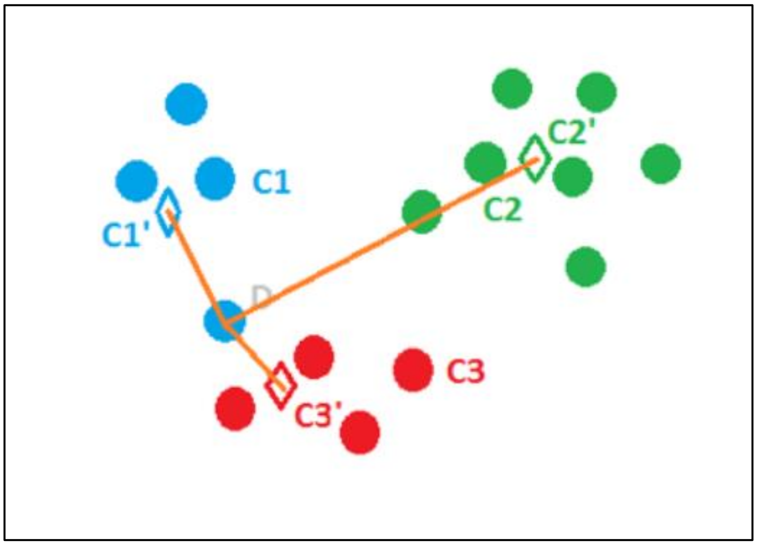
μ_K is the new centroid

The significance of *argmin* in assignment step equation is as follows:

For a i^{th} data point which is a 2d object and μ which is again a 2d object (Centroid), we compute the distance between these two, this is given by $d(X_i, \mu_K)$, where k is the number of clusters and then from these k different results we will choose the minimum of all.

Step three: *Revise cluster centers as mean of assigned observations*

Now we have assigned all the points based on which cluster center they were closest to. Next, we need to update the cluster centers based on the points assigned to them. For instance, we can find the center mass of the blue cluster by summing over all the blue points and dividing by the total number of points, which is four here. And the resulted center mass C1', represented by a blue diamond, is our new center for the blue cluster. Similarly, we can find the new centers C2' and C3' for the green and red clusters.



This step is called ***optimization step*** as in the step, the algorithm calculates the average of all the points in a cluster and moves the centroid to that average location.

The **equation for optimization step** is as follows:

$$\mu_k = \frac{1}{n_k} \sum_{i:z_i=k} X_i$$

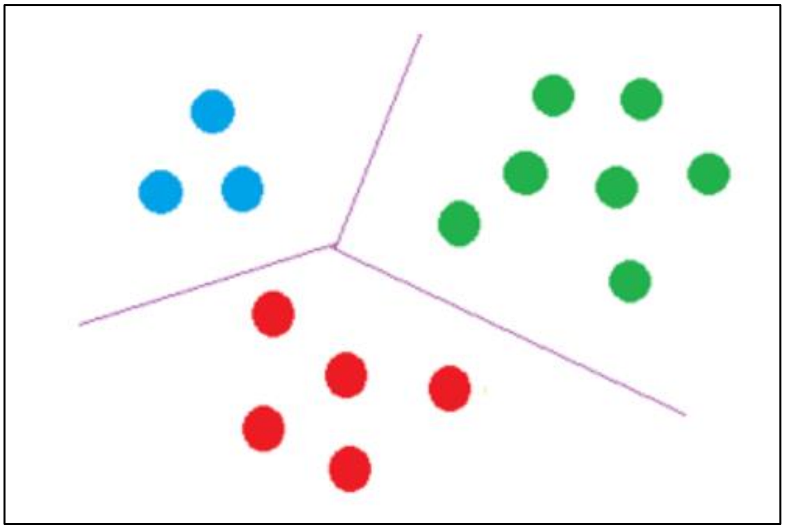
Where, X_i is the i^{th} data point

μ_K is the new centroid

n_K is number of data points with in a cluster

Step four: *Repeat step 2 and step 3 until convergence*

The last step of k-means is just to repeat the above two steps. For example, in this case, once C1’, C2’ and C3’ are assigned as the new cluster centers, point D becomes closer to C3’ and thus can be assigned to the red cluster. We keep on iterating between assigning points to cluster centers and updating the cluster centers until **convergence**. Finally, we may get a solution like the following figure.



The cost function for the K-Means algorithm is given as:

$$J = \sum_{i=1}^{n} ||X_i - \mu_{k(i)}||^2 = \sum_{k=1}^{K} \sum_{i \epsilon C_k} ||X_i - \mu_k||^2$$

Where, X_i is the i^{th} data point

μ_K is the new centroid

c) How is the value of ‘k’ chosen in K-means clustering? Explain both the statistical as well as the business aspect of it.

The K-Means algorithm is simple and most commonly used algorithm for clustering.

The basic idea behind k-means consists of defining k clusters such that total **intra-cluster variation (or error) is minimum and inter-cluster variation is maximum.**

A cluster center is the representative of its cluster. The squared distance between each point and its cluster center is the required variation. **The aim of k-means clustering is to find these k clusters and their centers while reducing the total error.**

Below two statistical methods are useful to find this k in k-Means.

These methods are:

1. **The Elbow Method**
2. **The Silhouette Method**

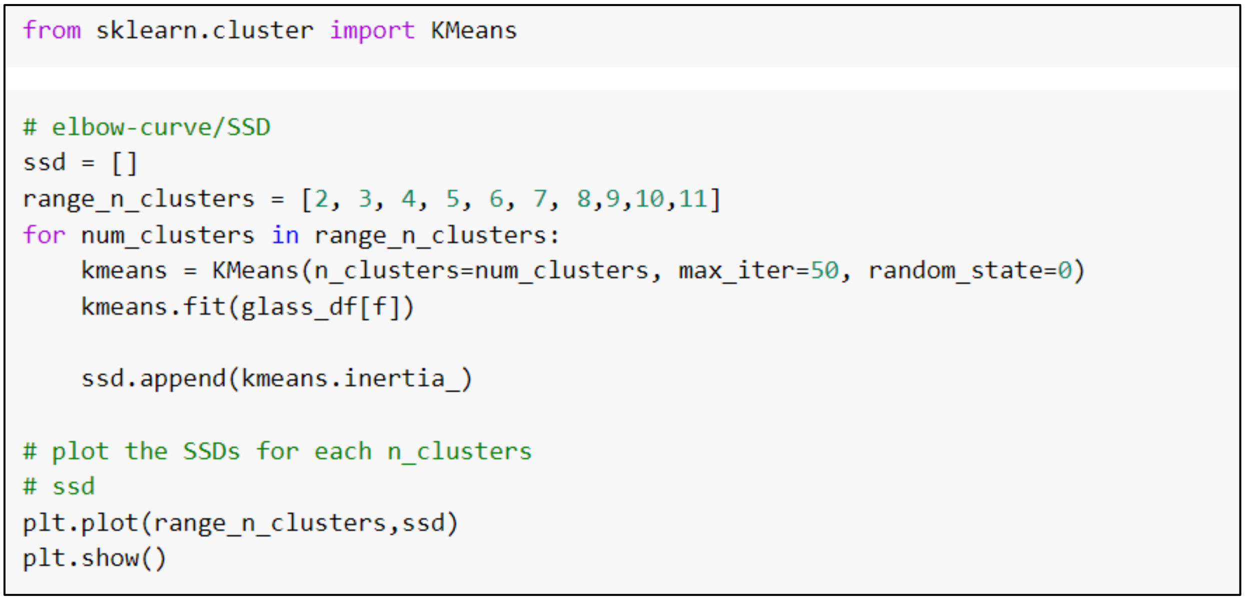
The Elbow Method: This is probably the most well-known method for determining the optimal number of clusters.

Here we calculate the **Within-Cluster-Sum of Squared** Errors (WSS) for **different values of k** and choose the k for which WSS becomes first starts to **diminish**. In the plot of WSS-versus-k, this is visible as an **elbow.**

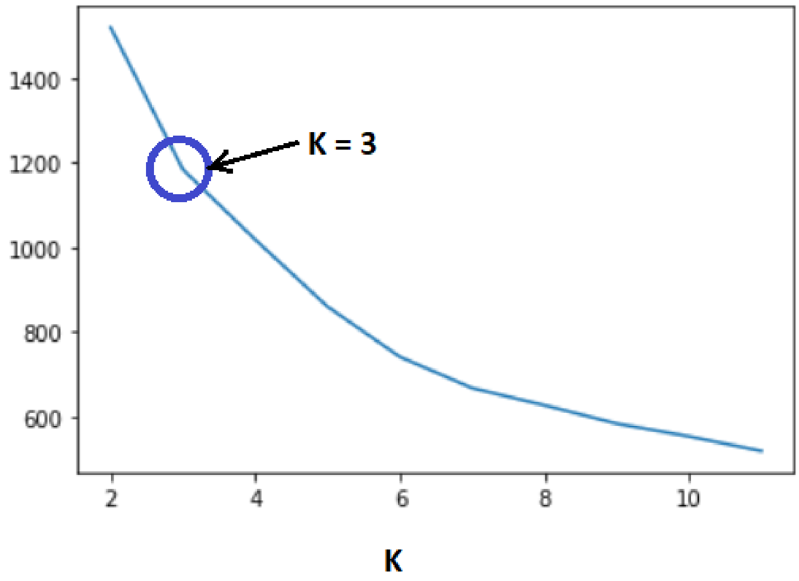
Within-Cluster-Sum of Squared Errors (WSS) is defined as below:

- The **Squared Error for each point** is the square of the distance of the point from its representation i.e. its predicted cluster center.
- The WSS score is the **sum of these Squared Errors for all the points**.
- Any distance metric like the **Euclidean Distance** or the **Manhattan Distance** can be used.

The Elbow Curve can be easily calculated in Python using the metrics module of the ***sklearn*** library.



```
from sklearn.cluster import KMeans


# elbow-curve/SSD
ssd = []
range_n_clusters = [2, 3, 4, 5, 6, 7, 8,9,10,11]
for num_clusters in range_n_clusters:
    kmeans = KMeans(n_clusters=num_clusters, max_iter=50, random_state=0)
    kmeans.fit(glass_df[f])

    ssd.append(kmeans.inertia_)

# plot the SSDs for each n_clusters
# ssd
plt.plot(range_n_clusters,ssd)
plt.show()
```



The above **plot looks like an arm with a clear elbow at k = 3.**

The Silhouette Method: The silhouette value measures how similar a point is to its own cluster (**cohesion**) compared to other clusters (**separation**).

The range of the Silhouette value is between +1 and -1. A **high value is desirable** and indicates that the point is placed in the correct cluster. If many points have a negative Silhouette value, it may indicate that we have created too many or too few clusters.

The Silhouette Value ***s(i)*** for each data point *i* is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_i| > 1$$

and

$$s(i) = 0, \text{ if } |C_i| = 1$$

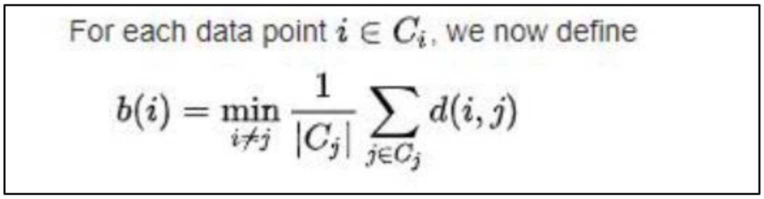
s(i) is defined to be equal to zero if *i* is the only point in the cluster. This is to prevent the number of clusters from increasing significantly with many single-point clusters.

Here, ***a(i)*** is the measure of similarity of the point *i* to its own cluster. It is measured as the average distance of *i* from other points in the cluster.

For each data point $i \in C_i$ (data point i in the cluster C_i), let

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

Similarly, ***b(i)*** is the measure of dissimilarity of *i* from points in other clusters.

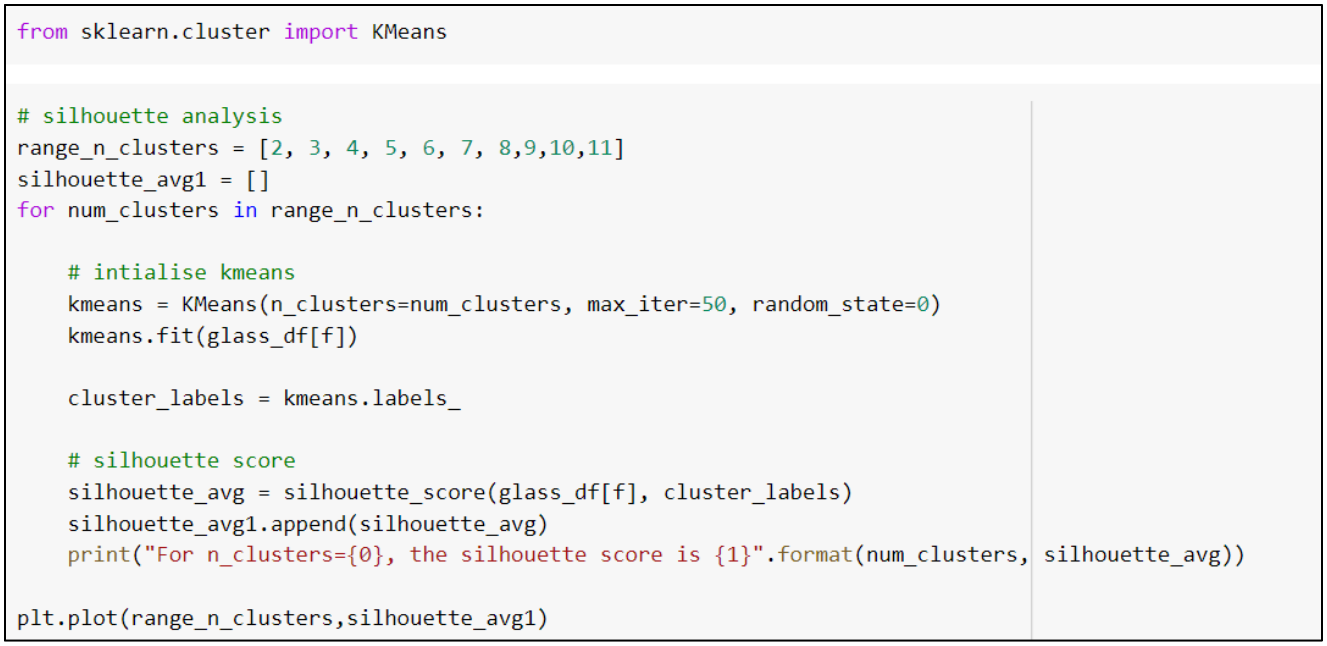


For each data point $i \in C_i$, we now define

$$b(i) = \min_{i \neq j} \frac{1}{|C_j|} \sum_{j \in C_j} d(i, j)$$

d(i, j) is the distance between points *i* and *j*. Generally, **Euclidean Distance** is used as the distance metric.

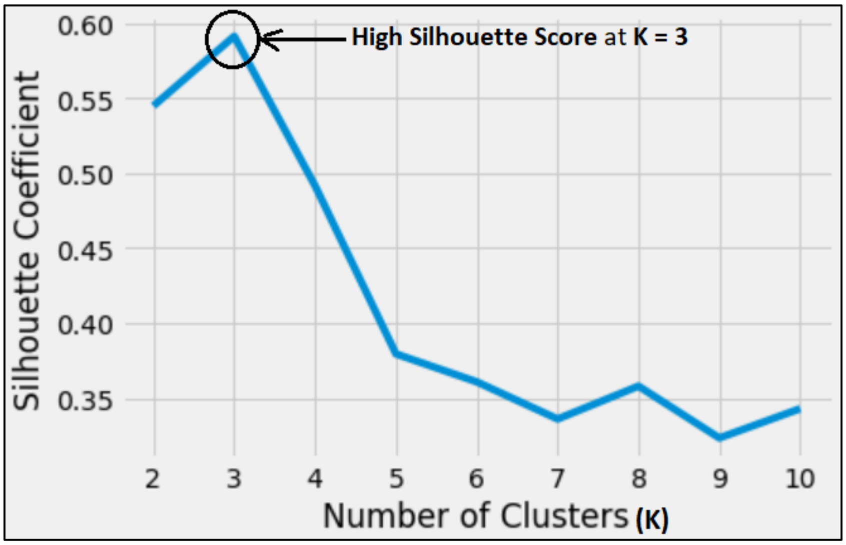
The Silhouette score can be easily calculated in Python using the metrics module of the ***sklearn*** library.



```
from sklearn.cluster import KMeans

# silhouette analysis
range_n_clusters = [2, 3, 4, 5, 6, 7, 8,9,10,11]
silhouette_avg1 = []
for num_clusters in range_n_clusters:

    # intialise kmeans
    kmeans = KMeans(n_clusters=num_clusters, max_iter=50, random_state=0)
    kmeans.fit(glass_df[f])

    cluster_labels = kmeans.labels_

    # silhouette score
    silhouette_avg = silhouette_score(glass_df[f], cluster_labels)
    silhouette_avg1.append(silhouette_avg)
    print("For n_clusters={0}, the silhouette score is {1}".format(num_clusters, silhouette_avg))

plt.plot(range_n_clusters,silhouette_avg1)
```

As we know a **high Silhouette Score** is desirable. The Silhouette Score reaches its ***global maximum at the optimal k***. This should ideally appear as a peak in the Silhouette Value-versus-k plot.

Here is the plot for our dataset:



The Elbow Method is more of a **decision rule**, while the **Silhouette is a metric** used for validation while clustering. Thus, it can be used in combination with the Elbow Method.

Therefore, the Elbow Method and the Silhouette Method are not alternatives to each other for finding the optimal K. **Rather they are tools to be used together for a more confident decision**.

Business Aspect:

Business aspect of deciding k means how many numbers of cluster 'k' are needed to choose with respect to business domain understanding. Cluster analysis is a technique used in machine learning which groups data points together based on the similarities between them. One can use various clustering algorithms which provide valuable insights about your business. The information generated from clustering can be used across your business functions to create a profitable consumer response.

As far as clustering algorithms go, it is simple and flexible to use in retail business.
It needs to specify the number of clusters, which can be time-consuming or detrimental to business if they don't follow a statistical or knowledge-backed method.

Working with the optimal number of clusters for retail data and market environment will facilitate the use of resources in a more efficient and effective manner. One can select the number of clusters using industry-related knowledge or different statistical methods that is already discussed above.

Below are some of the business aspect based on which optimal number of clusters can be chosen:

- We can use demographic, psychographic and behavioral data as well as performance data to cluster the consumers for a product category. This is a part of consumer segmentation.

- The delivery routes and patterns of trucks and drones have been monitored to find the optimal launch locations, routes and destinations for the company. This is a part of delivery optimization.

- You can use variables such as frequency of purchases, how recently the consumer visited the store, average spend per trip and basket composition to analyses and predict retention rates of customer segments, clustering or segmentation can be done based on RFM (recency, frequency, monetary value) analysis. This is a part of customer retention.

Our goal shouldn't be to just create clusters from the data. It should be to create meaningful, accurate clusters that business can use to apply new rule or strategy.

d) Explain the necessity for scaling/standardization before performing Clustering.

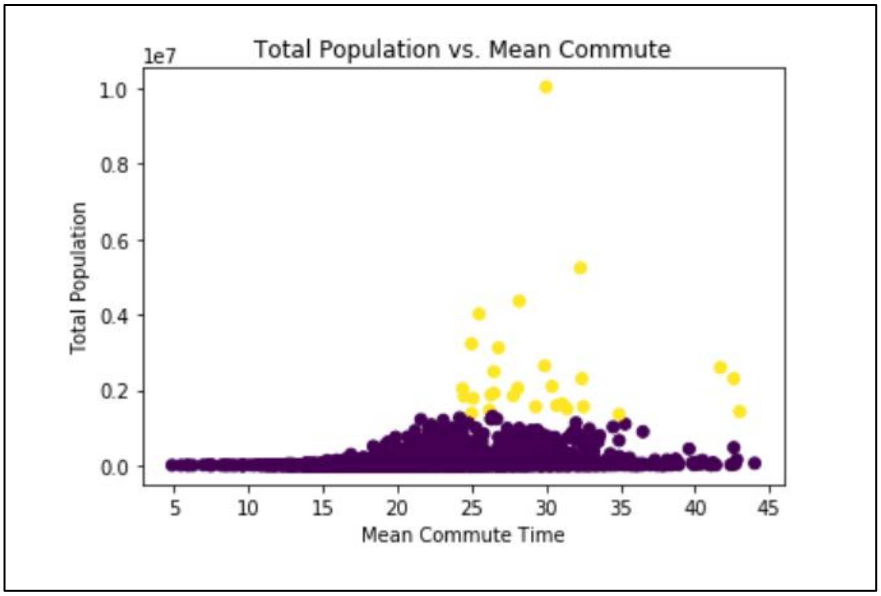
In statistics, **standardization** (sometimes called data normalization or feature scaling) refers to the process of rescaling the values of the variables in analysis data set so they share a **common scale**.

Often performed as a pre-processing step, particularly for cluster analysis, standardization may be important if you are working with data where each variable has a different unit (e.g., inches, meters, tons and kilograms), or where the scales of each of your variables are very different from one another (e.g., 0-1 vs 0-1000). The reason this importance is particularly high in cluster analysis is because groups are defined based on the distance between points in mathematical space.

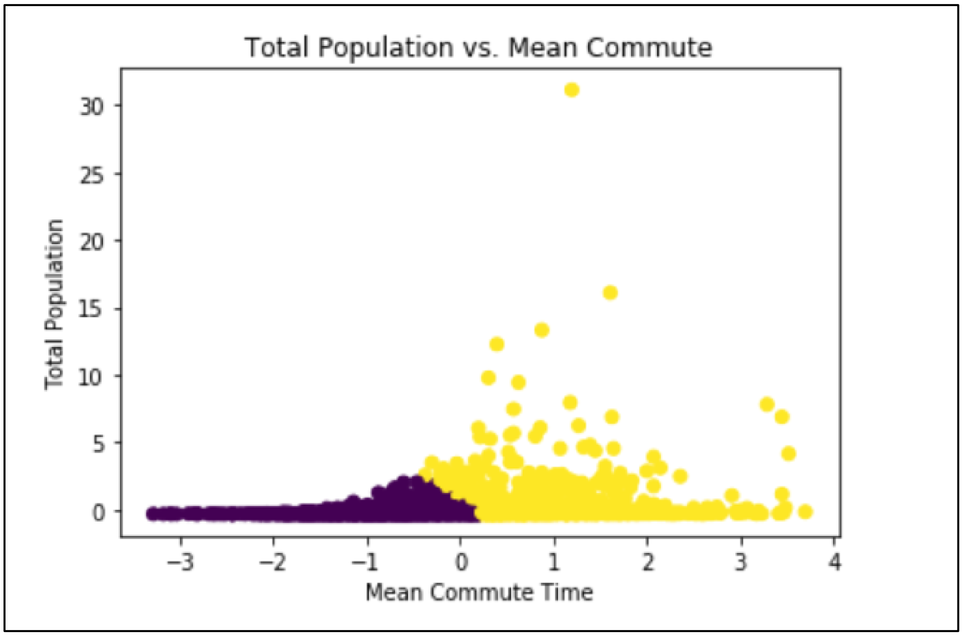
In a situation where one field has a much greater range of value than another (because the field with the wider range of values likely has greater distances between values), it may end up being the primary driver of what defines clusters. Standardization helps to make the relative weight of each variable equal by converting each variable to a unitless measure or relative distance.

Let’s take an example of US census demographic data. We are performing cluster analysis on Total Population and Mean Commute Time. The units (number of people vs. minutes) and the range of values (85 - 10038388 people vs. 5 - 45 minutes) of these attributes are very different.

When we create clusters with the raw data, we see that Total Population is the primary driver of dividing these two groups.



However, after standardization, both Total Population and Mean Commute seem to have an influence on how the clusters are defined.



Hence, we can see that, standardization prevents variables with larger scales from dominating how clusters are defined. It allows all variables to be considered by the algorithm with equal importance.

e) Explain the different linkages used in Hierarchical Clustering.

Let’s first explain the concept of linkage in Hierarchical Clustering. The process of Hierarchical Clustering involves either clustering sub-clusters (data points in the first iteration) into larger clusters in a bottom-up manner or dividing a larger cluster into smaller sub-clusters in a top-down manner. During both the types of hierarchical clustering, the distance between two sub-clusters needs to be computed.

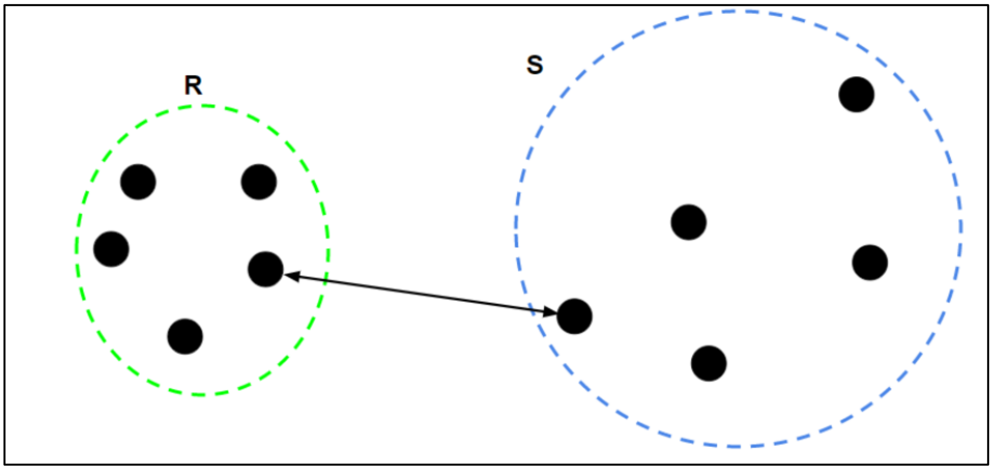
The different types of linkages used in hierarchical clustering are:

- Single Linkage
- Complete Linkage
- Average Linkage

Single Linkage :

In single linkage hierarchical clustering, the distance between two clusters is defined as ***the shortest distance between two points in each cluster***. For example, for two clusters R and S, the single linkage returns the **minimum** distance between two points i and j such that i belongs to R and j belongs to S.

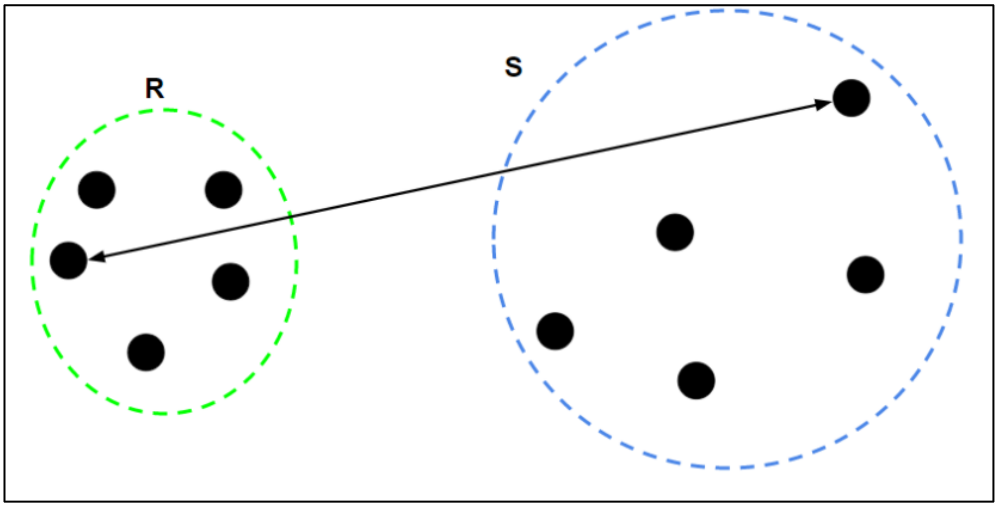
$$\boxed{L(R,S) = min(D(i,j)), i\epsilon R, j\epsilon S}$$



Complete Linkage:

In complete linkage hierarchical clustering, the distance between two clusters is defined ***as the longest distance between two points in each cluster.*** For example, for two clusters R and S, the complete linkage returns the **maximum** distance between two points i and j such that i belongs to R and j belongs to S.

$$\boxed{L(R,S) = max(D(i,j)), i\epsilon R, j\epsilon S}$$



Average Linkage:

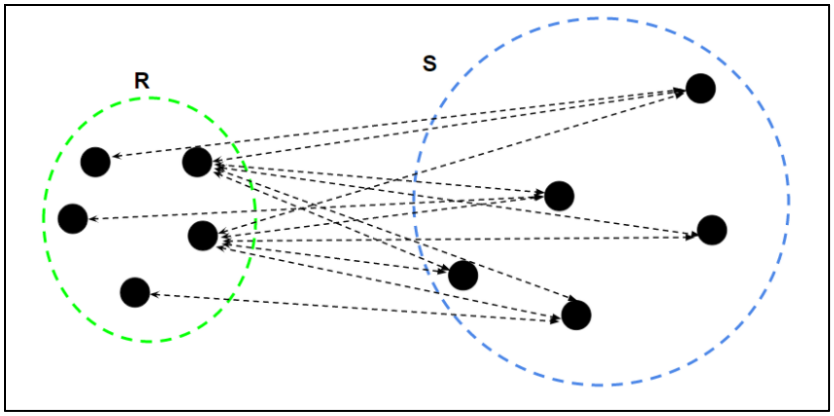
In average linkage hierarchical clustering, the distance between two clusters is defined as ***the average distance between each point in one cluster to every point in the other cluster***. For example, the distance between clusters R and S to the left is equal to the average length each arrow between connecting the points of one cluster to the other.

$$\boxed{L(R,S) = \frac{1}{n_R+n_S} \sum_{i=1}^{n_R} \sum_{j=1}^{n_S} D(i,j), i\epsilon R, j\epsilon S}$$

where

n_R – Number of data-points in R

n_S – Number of data-points in S



We should decide what type of linkage should be used by looking at the data. One convenient way to decide is to look at how the dendrogram looks. Usually, **single linkage type will produce dendrograms which are not structured properly**, whereas **complete or average linkage** will produce clusters which have a proper **tree-like structure**.

------END------