1. For Clustering N points we need N iteration and for each iteration we need to find out the minimum distance by comparing their distances and its given the time complexity to compute distance is O( and for N iterations we have N times complexity of each iteration . Hence the complexity of Agglomerative Hierarchical Clustering is

Complexity = **O()** , where N = number of points

1. I have used Euclidean Distance to compute the distances and single-linkage is used to cluster the points . The python code for the same is as follows.

from sklearn.cluster import AgglomerativeClustering

import numpy as np

import matplotlib.pyplot as plt

#Getting the data ready

array\_txt = np.loadtxt(*r"C:\Users\gutti\Downloads\B.txt"*,usecols=(0, 1), skiprows=1)

df = array\_txt[:,0:2]

#Creating the model

agg\_clustering = AgglomerativeClustering(n\_clusters = 2, affinity = *'euclidean'*, linkage = *'single'*)

#predicting the labels

labels = agg\_clustering.fit\_predict(df)

#Plotting the results

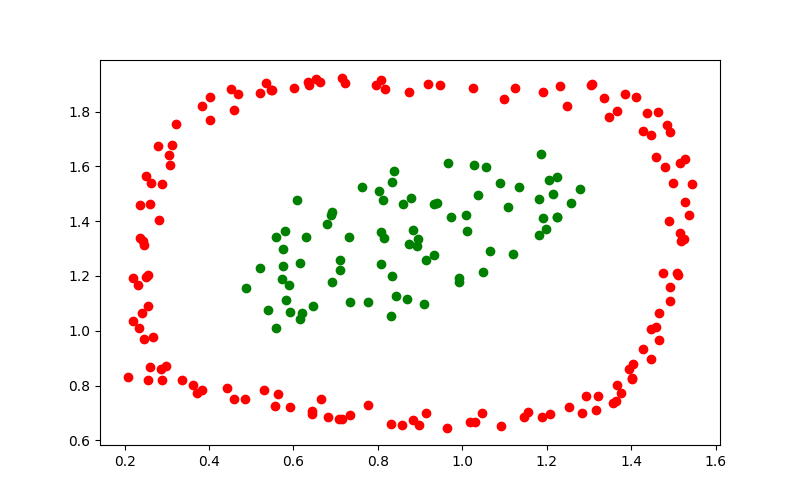
plt.figure(figsize = (8,5))

plt.scatter(df[labels == 0 ,0 ] ,df[labels == 0,1 ], c = *'red'*)

plt.scatter(df[labels == 1 ,0] ,df[labels == 1,1 ] , c = *'green'*)

plt.show()

The output of the code above and the resultant clusters are



Where green is cluster1 and red is cluster 2