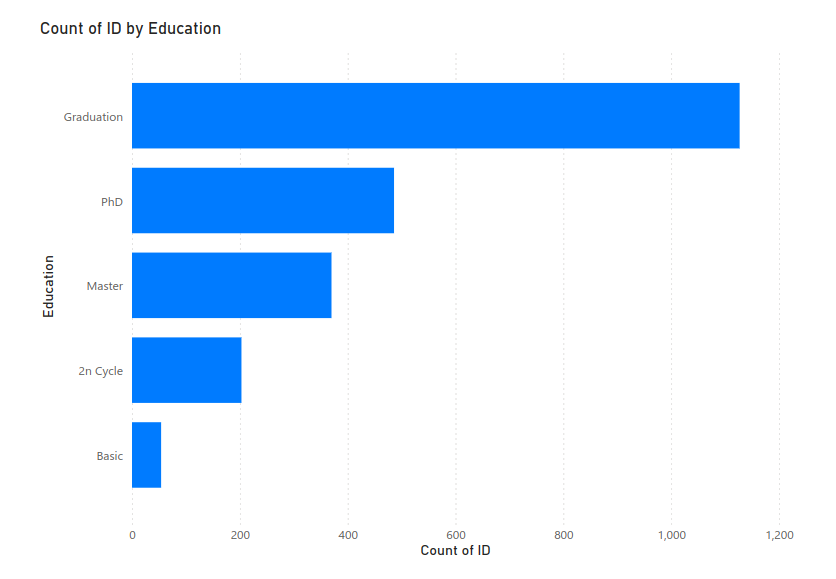
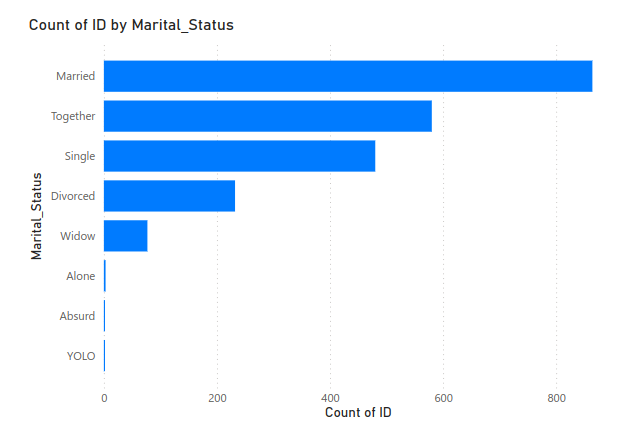
***PART 1***

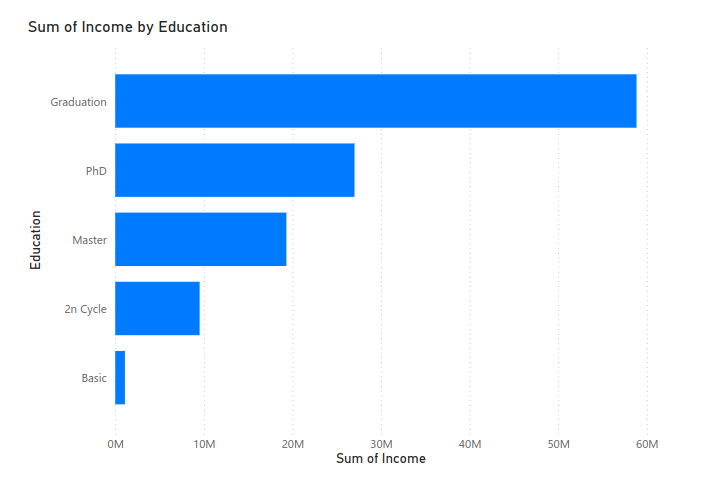
Bar Graph Visualizations:

******

The bar graph indicates that the majority of customers have a "Graduation" level of education, with over 1,000 individuals falling into this category. In contrast, the number of customers with a "Basic" education is the smallest, highlighting a significant disparity in educational distribution.

******

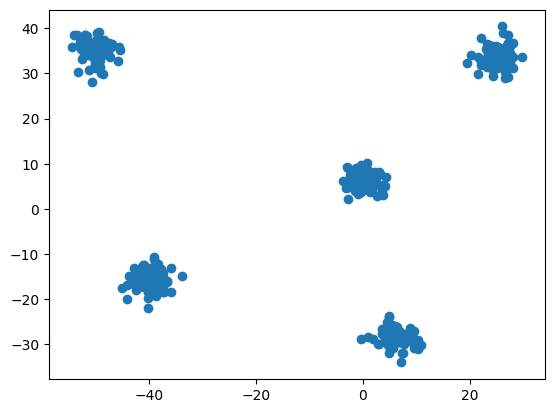
The bar graph indicates that more than 1400 customers are married or together and only around 800 are single individuals. This shows the trend that more couples in costumers

******

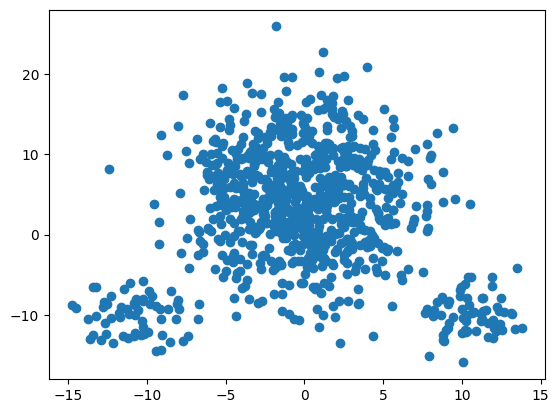
The bar graph shows that customers with a "Graduation" level of education have the highest total income, significantly more than the other education levels. In contrast, those with "Basic" education have the lowest total income, indicating a potential correlation between higher education and increased income.

***PART 3***

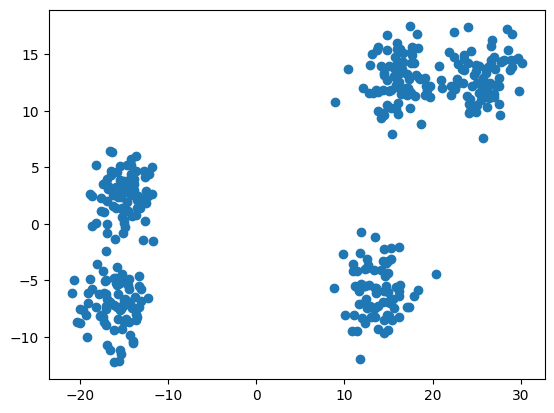
plt.scatter(Compact["x1"], Compact["x2"])



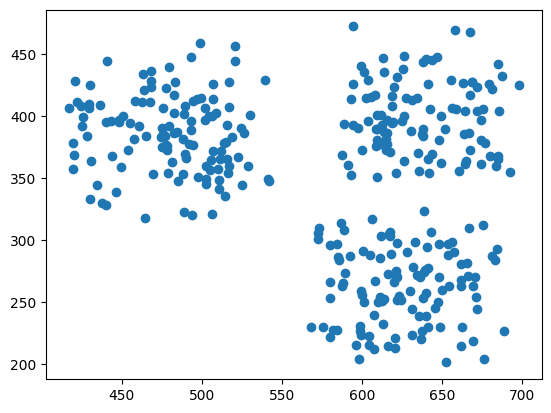
plt.scatter(Skewed["x1"], Skewed["x2"])

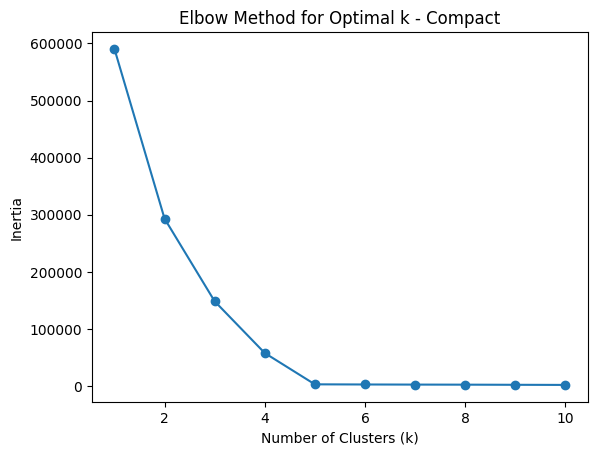


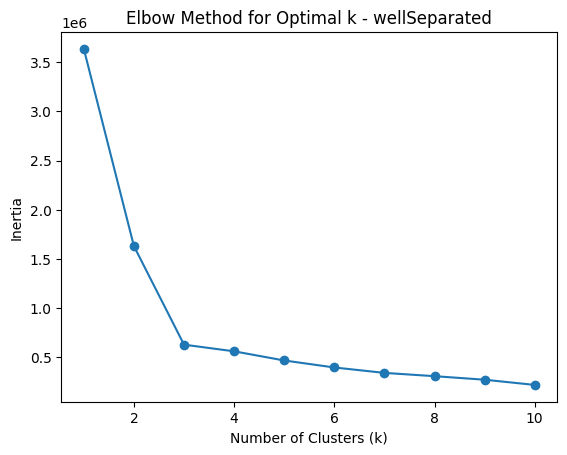
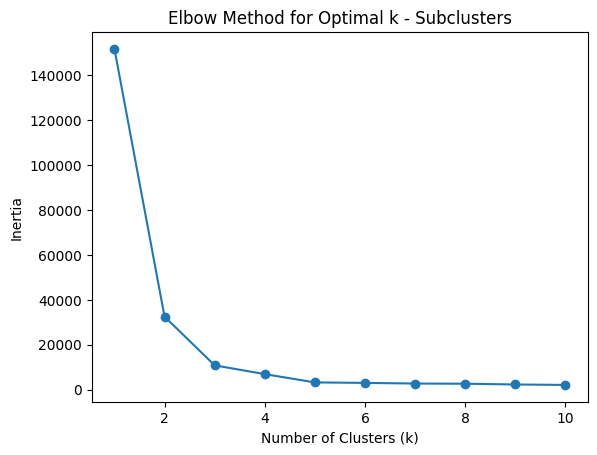
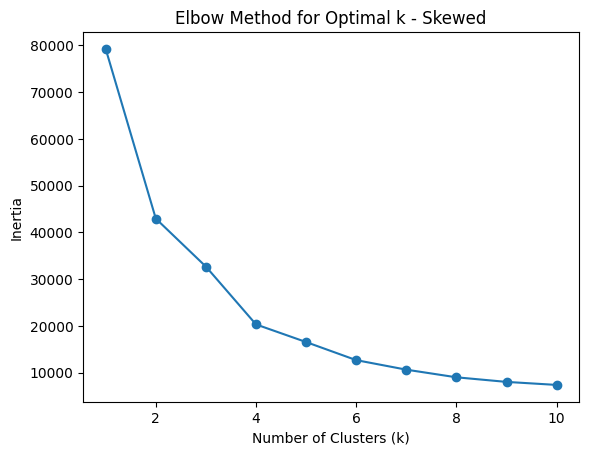
plt.scatter(Subclusters["x1"], Subclusters["x2"])



plt.scatter(wellSeperated["x1"], wellSeperated["x2"])

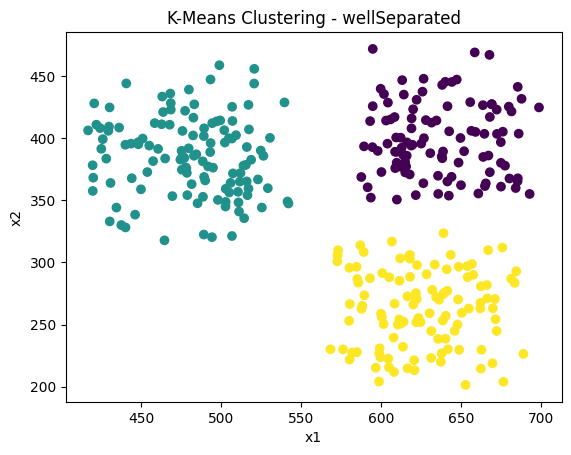
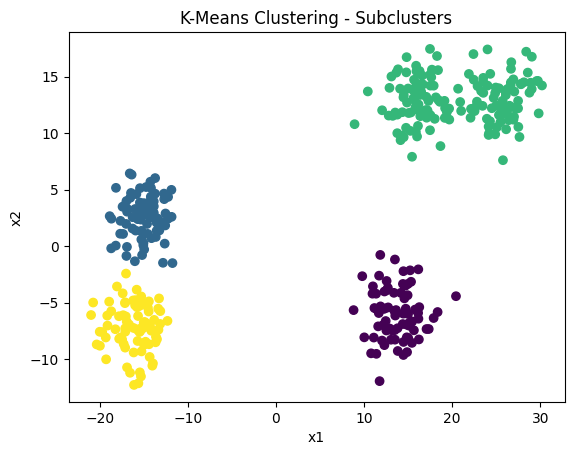
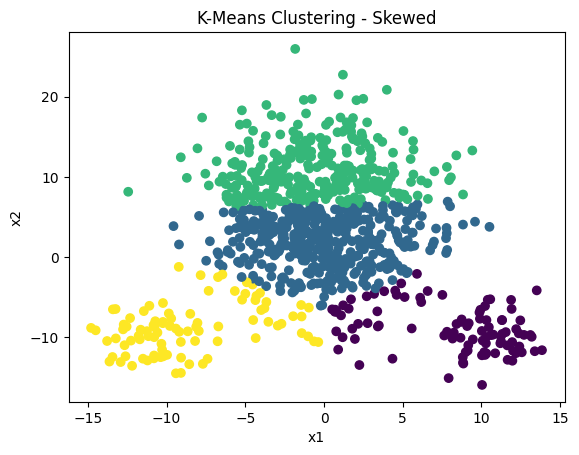
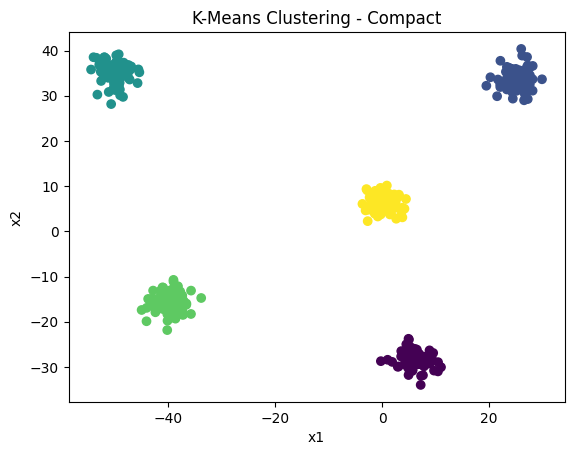


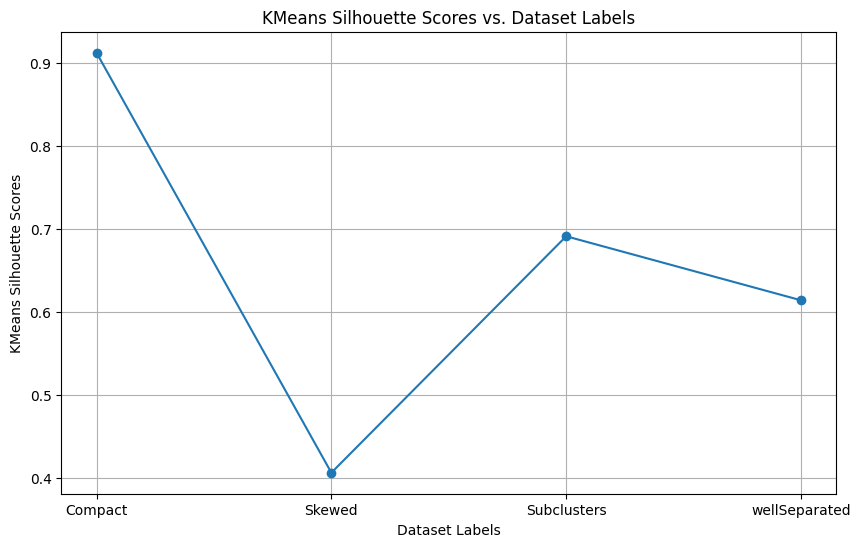
Elbow curves:  




datasets = [Compact, Skewed, Subclusters, wellSeperated]

optimal\_clusters\_per\_dataset = [5, 4, 4, 3]





KMeans Clustering (Compact Data):

Clusters are spherical and equally sized so it is performing optimally.

KMeans Clustering (Skewed Data):

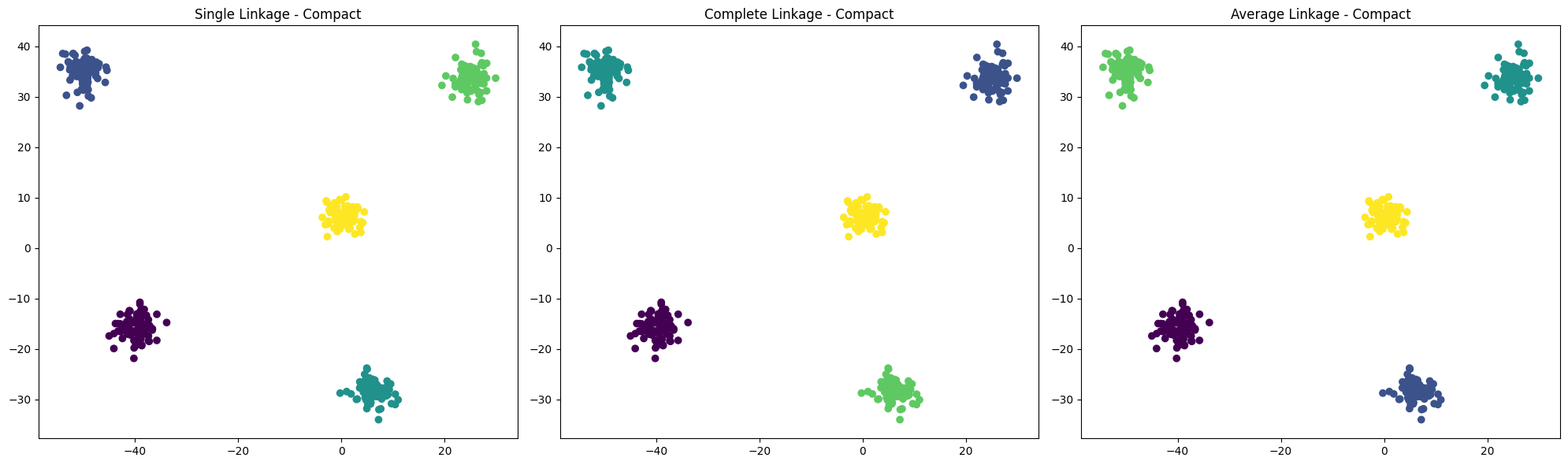
The algorithm fails in this case primarily due to its assumption that clusters are spherical and equally sized.

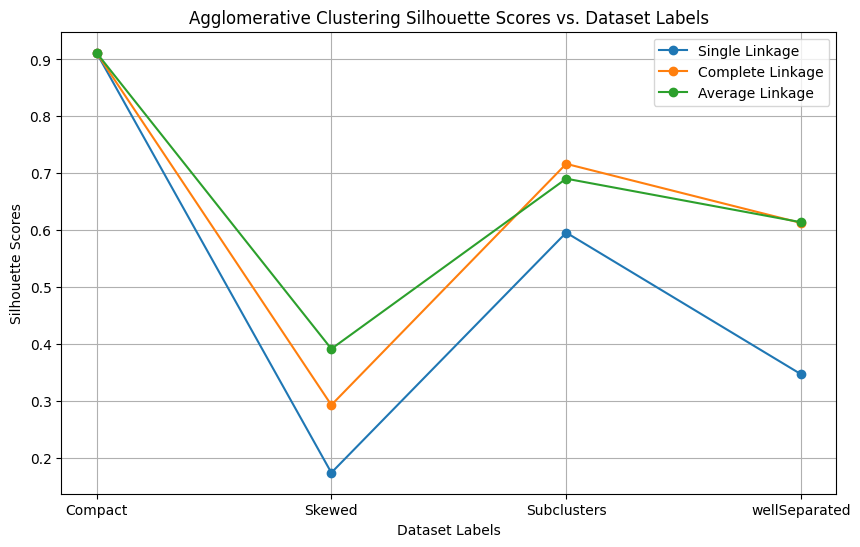
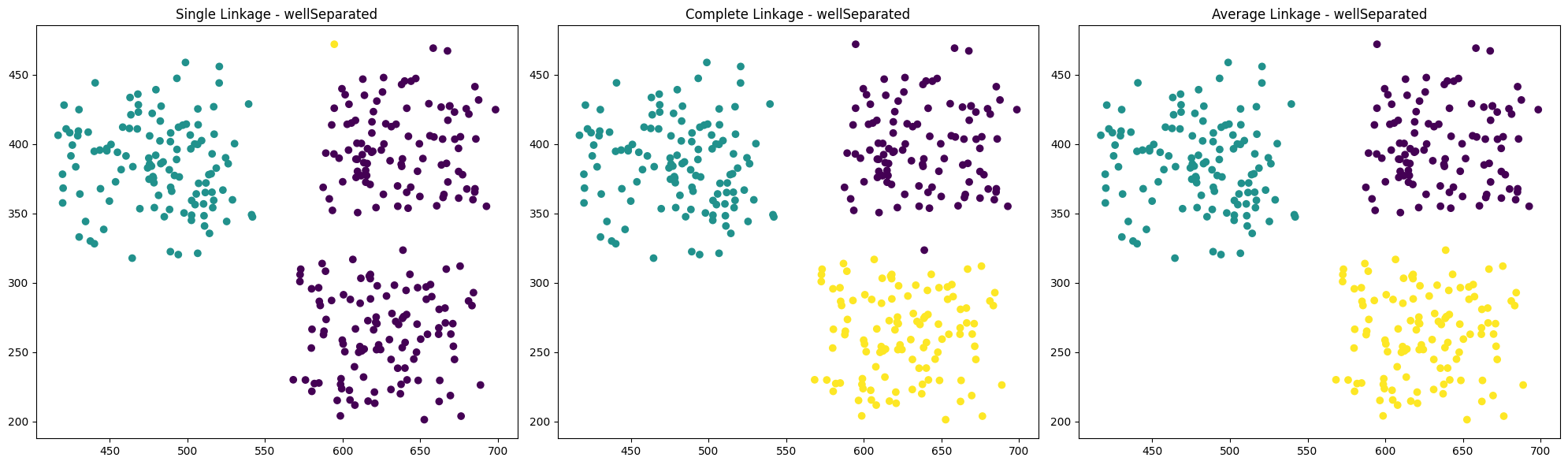
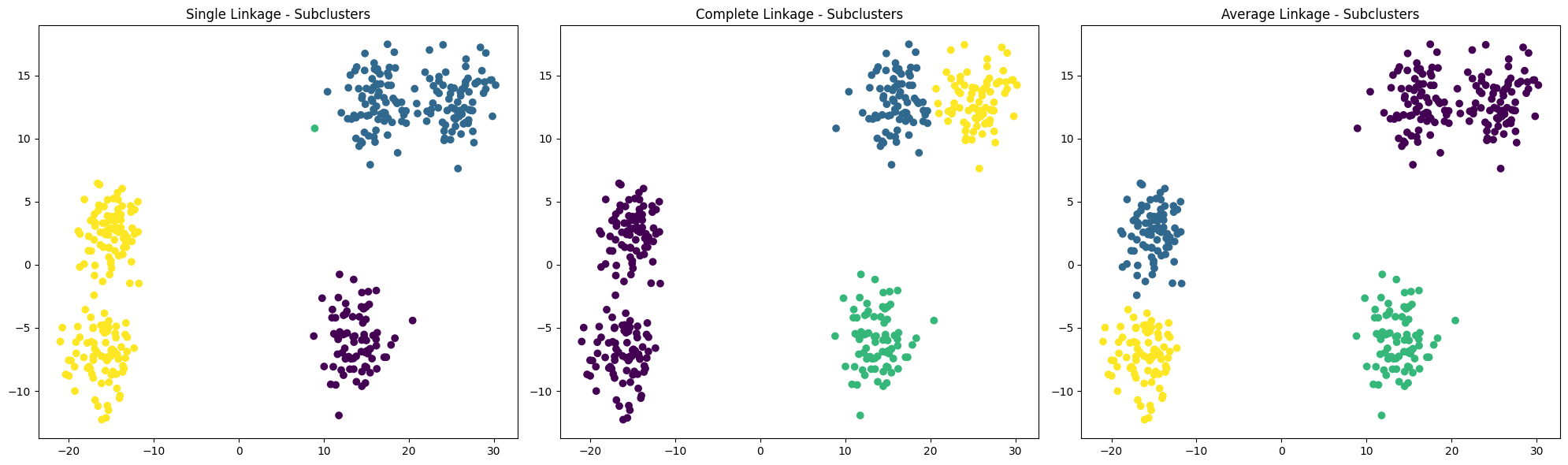
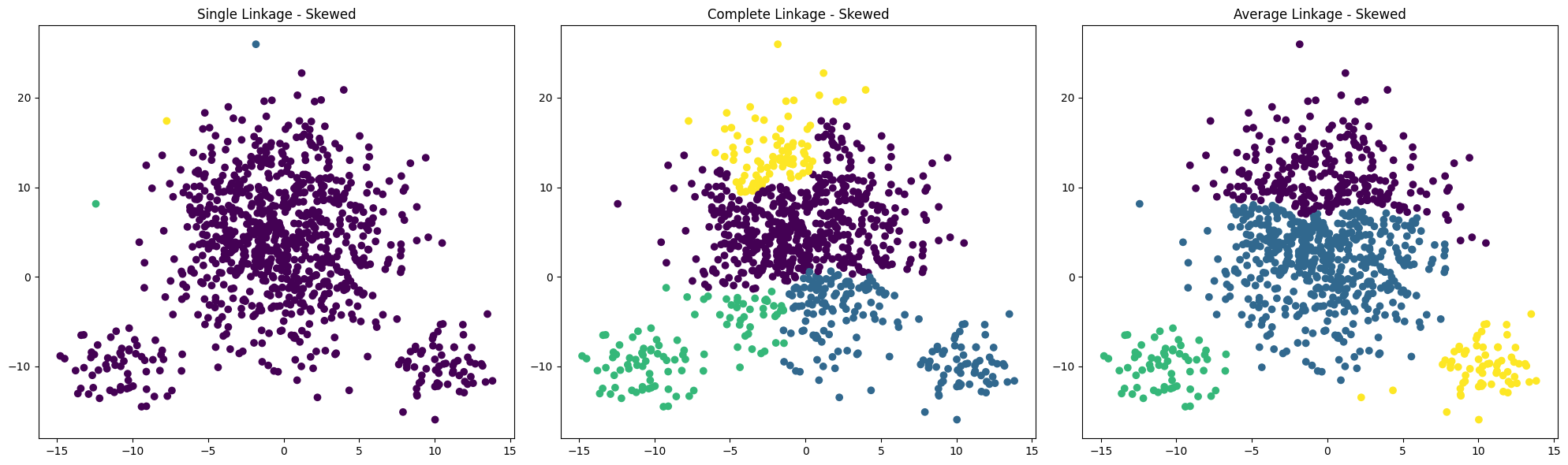
KMeans Clustering (SubClusters Data):

It assigns points to the nearest centroid, some more clusters could be created but arent, so it is performing sub-optimally.

KMeans Clustering (wellSeperated Data):

Works well as clusters are well-separated and have roughly equal sizes.





Aggloromative Clustering (Compact Data):

Works well especially with complete or average linkage, which consider the maximum or average distance between clusters, respectively.

Aggloromative Clustering (Skewed Data):

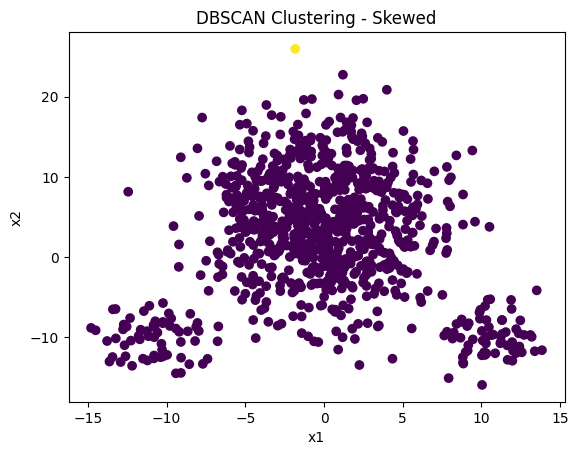
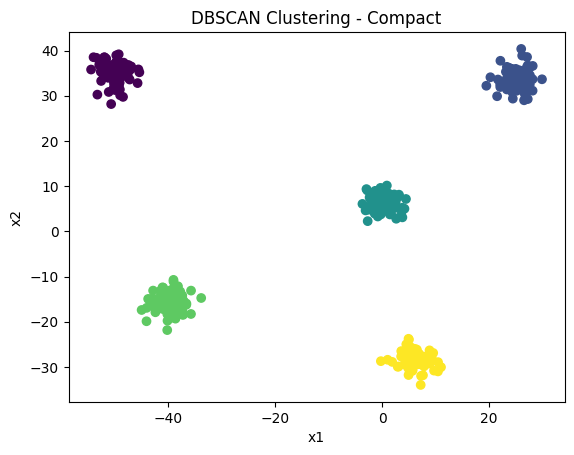
Similar to KMeans, it struggles with skewed clusters.

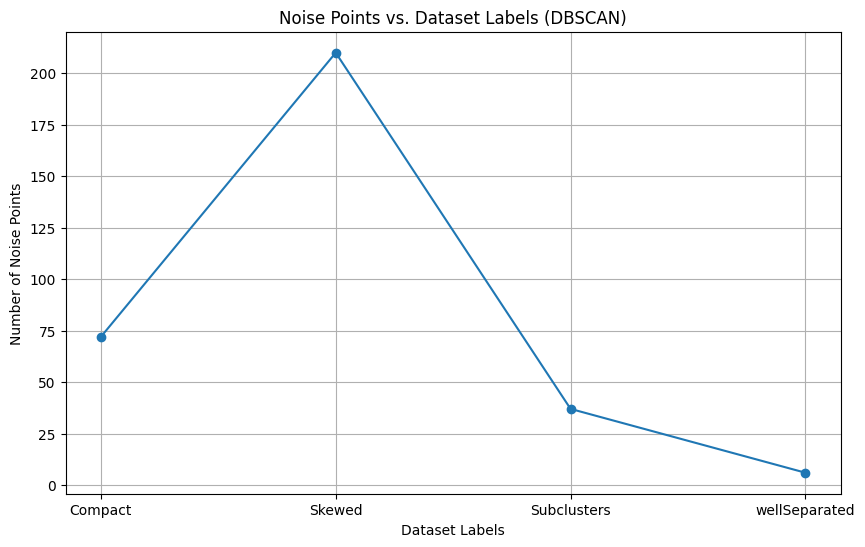
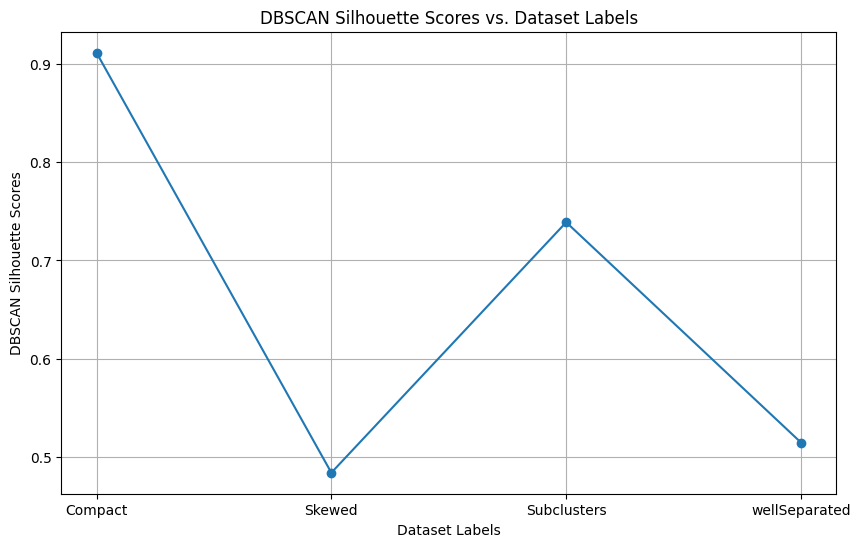
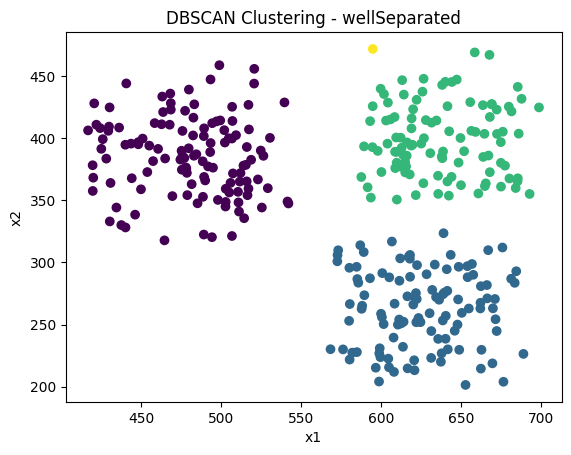
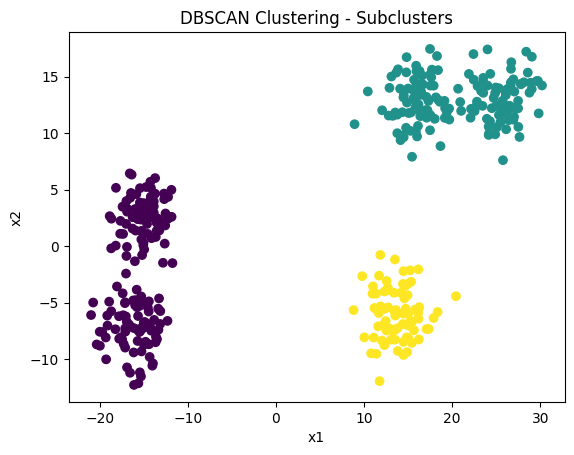
Aggloromative Clustering (SubClusters Data):

Works well especially if linkage strategies consider the average or complete distance between clusters.

Aggloromative Clustering (wellSeperated Data):

Performs well for well-separated clusters, especially with single or complete linkage.



DBSCAN Clustering (Compact Data):

Works well as the compact clusters have consistent densities.

DBSCAN Clustering (Skewed Data):

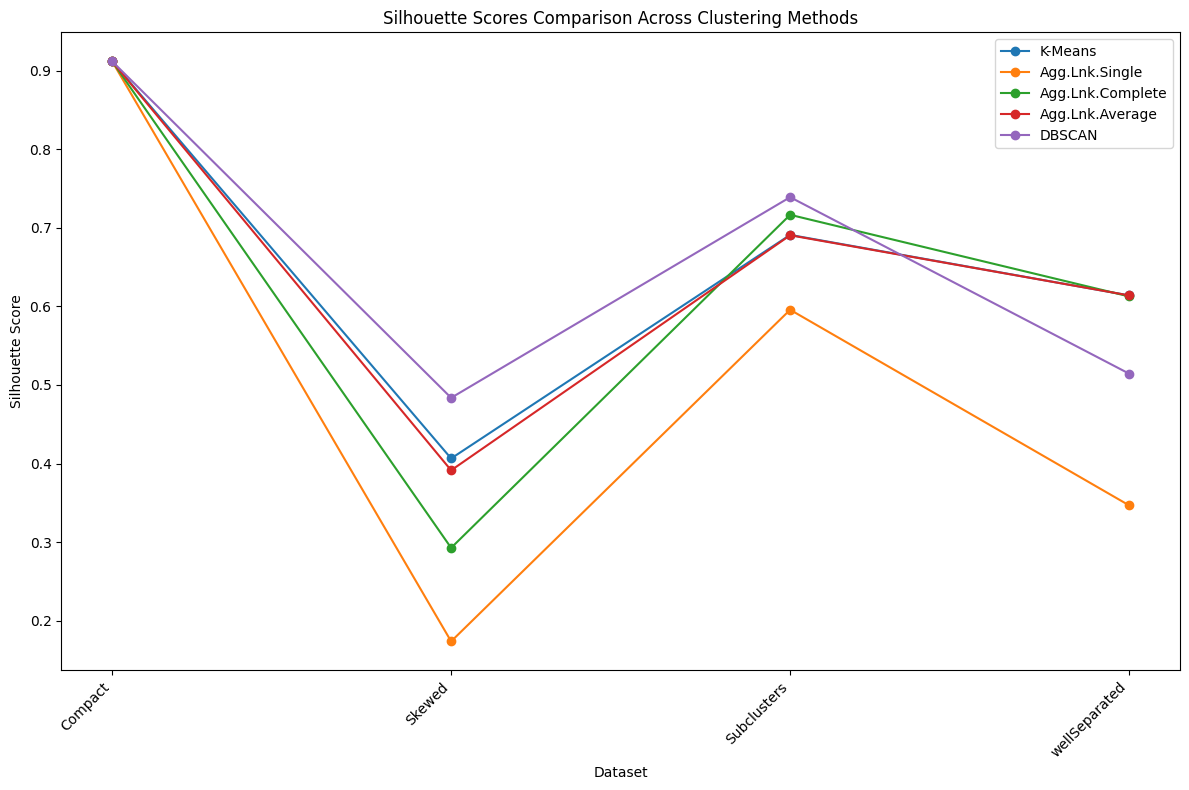
Works well and the algorithm can handle skewed clusters better than KMeans, as it doesn't assume any specific shape.

DBSCAN Clustering (SubClusters Data):

Performs well as subclusters have sufficient density, and parameter tuning is crucial.

DBSCAN Clustering (wellSeperated): May not be the best choice for well-separated clusters but the density of points within each cluster is sufficient so it's working well. Tuning parameters is crucial.

|  | **Dataset** | **K-Means** | **Agg.Lnk.Single** | **Agg.Lnk.Complete** | **Agg.Lnk.Average** | **DBSCAN** | **DBSCANNoisePoints** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | Compact | 0.911780 | 0.911780 | 0.911780 | 0.911780 | 0.911780 | 0 |
| **1** | Skewed | 0.406510 | 0.173779 | 0.292751 | 0.391412 | 0.483642 | 10 |
| **2** | Subclusters | 0.690960 | 0.595697 | 0.716513 | 0.690470 | 0.738949 | 0 |
| **3** | wellSeparated | 0.613946 | 0.346804 | 0.612632 | 0.613946 | 0.514372 | 310 |

DBSCAN Algorithm performs the best for almost each and every data.