Mining of Massive Data

Clustering - Assignment 2

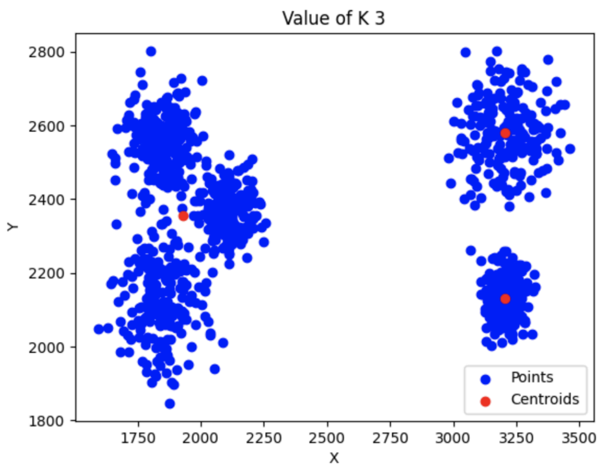
Analysis Report

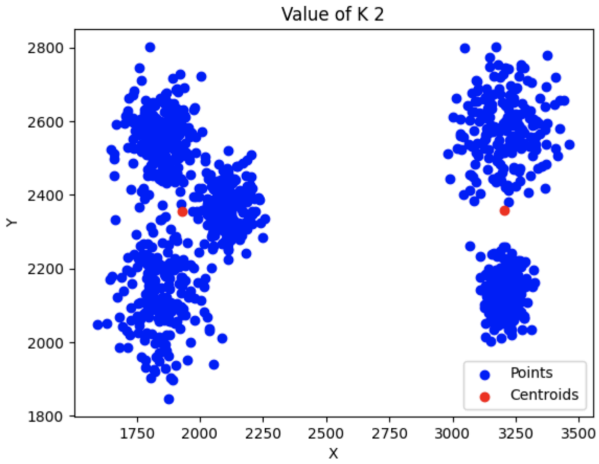
Kmeans Clustering using own Pyspark code on dataset DS1:

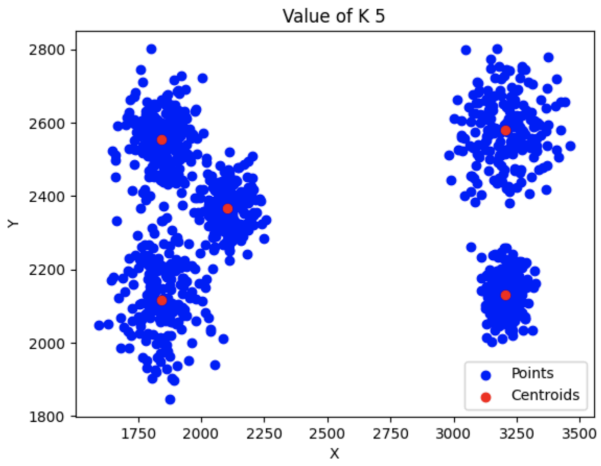
The analysis of K-means clustering reveals interesting trends across different values of K, shedding light on the optimal number of clusters for the dataset. As K increases, both the Within-Cluster Sum of Squared Errors (WSSE) and Between-Cluster Sum of Squared Errors (BSSE) decrease consistently. This trend suggests that higher values of K lead to better partitioning of the data, resulting in smaller within-cluster variances and larger between-cluster variances.

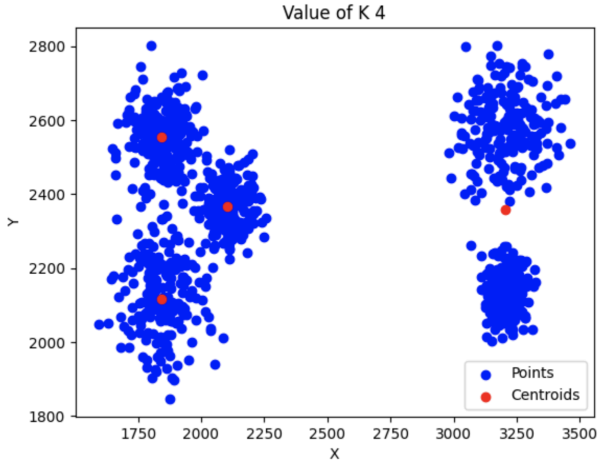
Notably, the most significant improvement in clustering quality is observed when transitioning from K=2 to K=3, indicating a notable enhancement in the cluster structure. Additionally, the Silhouette Coefficient (SC), which measures the compactness and separation of clusters, tends to improve with increasing K. K=5 emerges as the configuration yielding the highest silhouette coefficient, indicative of well-defined clusters with high cohesion and separation.

Despite variations in the final centers obtained from different runs of K-means clustering, the performance metrics such as WSSE, BSSE, and SC exhibit consistency across runs for each value of K. These findings collectively suggest that K=5 provides the optimal balance between cluster quality and computational efficiency for the dataset under consideration.









Custom Bisecting Kmeans Clustering on dataset DS1

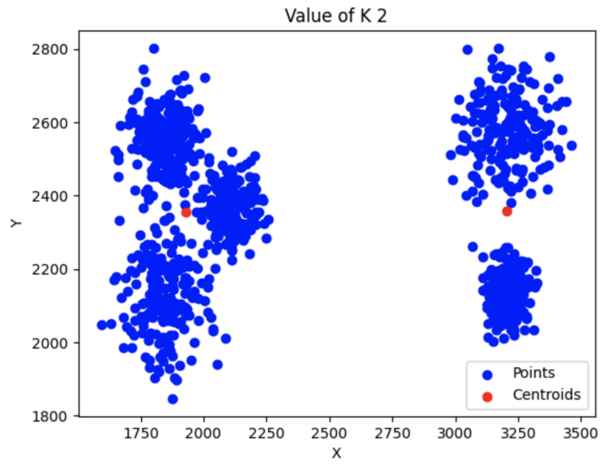
In the Custom Bisecting Kmeans Clustering analysis on dataset DS1, we explored various values of K (number of clusters) and evaluated the clustering performance using three metrics: Within-Cluster Sum of Squared Errors (WSSE), Between-Cluster Sum of Squared Errors (BSSE), and Silhouette Coefficient (SC).

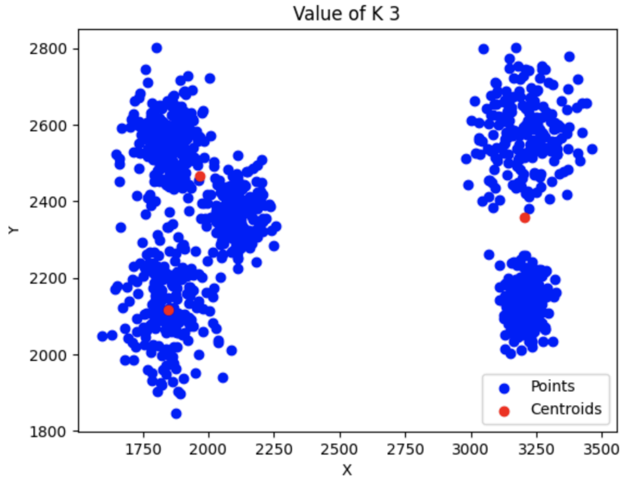
For K = 2, the average WSSE was notably high, indicating that the clustering may not have effectively captured the underlying structure of the dataset. The BSSE was also high, suggesting limited separation between clusters. However, the SC was relatively high at 0.739, indicating reasonable separation between clusters despite the high errors.

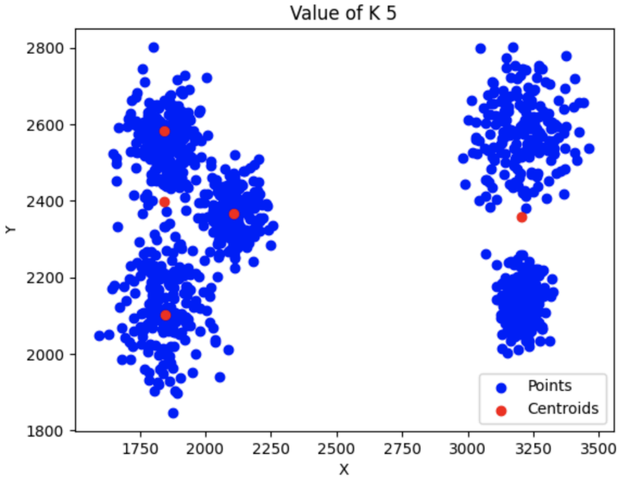
Increasing K to 3 resulted in a decrease in both WSSE and BSSE, suggesting better clustering performance in terms of data compression and cluster separation. However, the SC dropped substantially to 0.250, indicating less effective separation between clusters compared to K = 2.

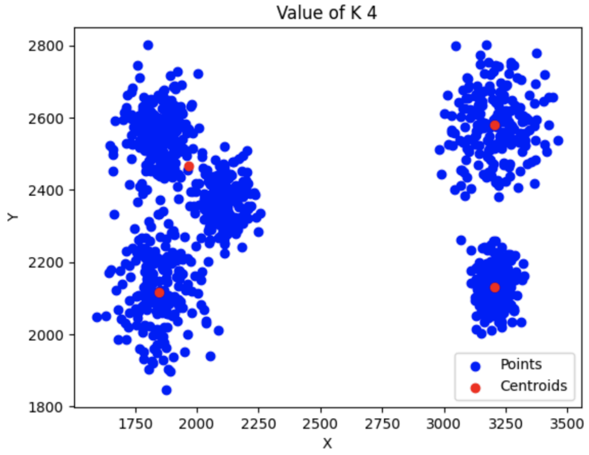
Further increasing K to 4 led to a slight decrease in WSSE and BSSE compared to K = 3, indicating continued improvement in clustering performance. However, the SC remained low at 0.224, suggesting that the additional cluster did not contribute significantly to better separation.

Finally, for K = 5, while there was a slight decrease in WSSE compared to K = 4, the BSSE did not decrease significantly. Moreover, the SC dropped further to 0.094, indicating poor separation between clusters.









K-MEANS clustering using PYSPARK MLLIB Kmeans function on DS2

The K-Means clustering algorithm was applied to dataset DS2 using PySpark MLlib's KMeans function with varying values of K. The performance of the algorithm was evaluated based on three key metrics: Within-Cluster Sum of Squared Errors (WSSE), Between-Cluster Sum of Squared Errors (BSSE), and Silhouette Coefficient (SC).

The results indicate a clear trend in the performance of the algorithm with different values of K. When K was set to 2, the average WSSE was found to be extremely high, suggesting poor clustering performance. However, the Silhouette Coefficient was notably high, indicating well-separated clusters despite the high within-cluster variance. The BSSE was relatively low, indicating some level of compactness in the clusters.

As K increased to 3, the average WSSE decreased significantly, indicating improved clustering performance in terms of within-cluster variance. The average BSSE also decreased, suggesting better separation between clusters. However, the Silhouette Coefficient decreased, indicating that the clusters were less well-separated compared to K=2.

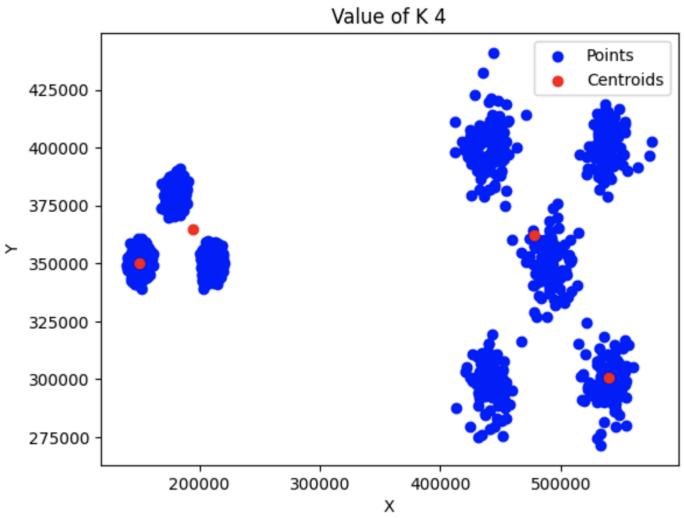
For K values of 4 and 5, the average WSSE and BSSE continued to decrease, indicating further improvement in clustering performance. However, the Silhouette Coefficient dropped significantly, suggesting that the clusters were becoming less well-separated with the increase in K.

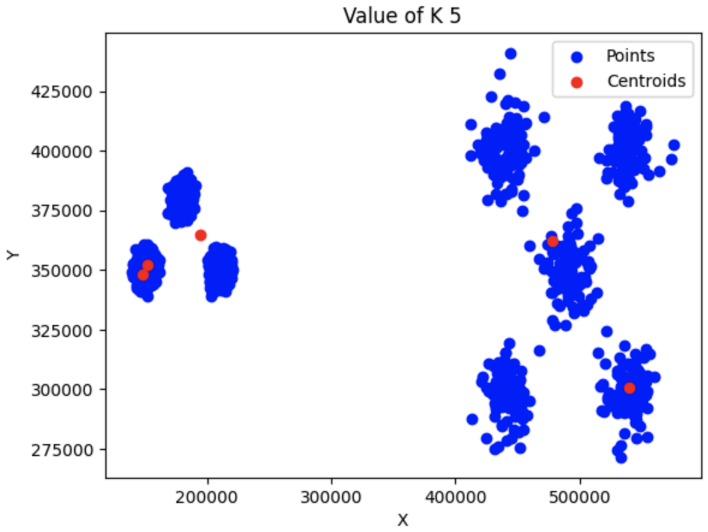
Interestingly, as K increased beyond 5, the average WSSE and BSSE continued to decrease, indicating further improvement in clustering performance in terms of compactness and separation. Moreover, the Silhouette Coefficient increased, suggesting that the clusters were becoming better-separated with the increase in K.

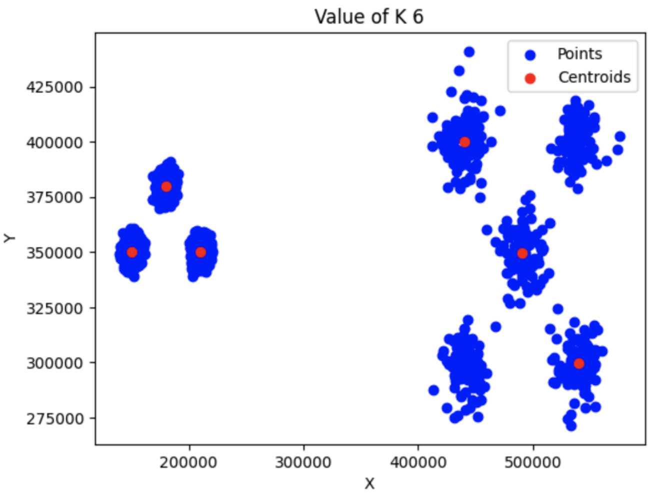
In conclusion, the analysis suggests that the K-Means clustering algorithm performs well on dataset DS2, with the optimal value of K being 8 based on the metrics evaluated. This value of K achieved a balance between within-cluster compactness and between-cluster separation, as indicated by the lowest average WSSE and BSSE, and the highest average Silhouette Coefficient among the tested values of K. However, it's essential to note that the performance of the algorithm may vary depending on the dataset and the specific characteristics of the data. Further exploration and tuning may be necessary to achieve the best clustering results for different datasets.
Results:

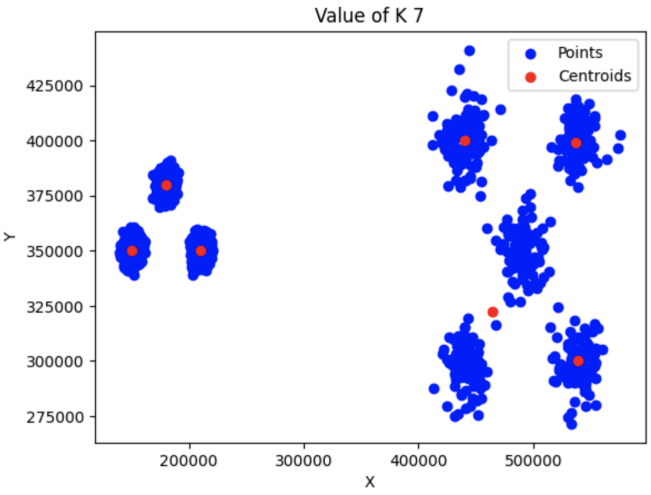
```
K = 2:
Avg WSSE: 7011360115400.669
Avg BSSE: 82578922701.47607
Avg SC: 0.977689319886371

K = 3:
Avg WSSE: 6164096499747.534
Avg BSSE: 164426221796.11627
Avg SC: 0.9429828430500716

K = 4:
Avg WSSE: 3404478608094.06
Avg BSSE: 194244817480.26083
Avg SC: 0.7075713508344729
```

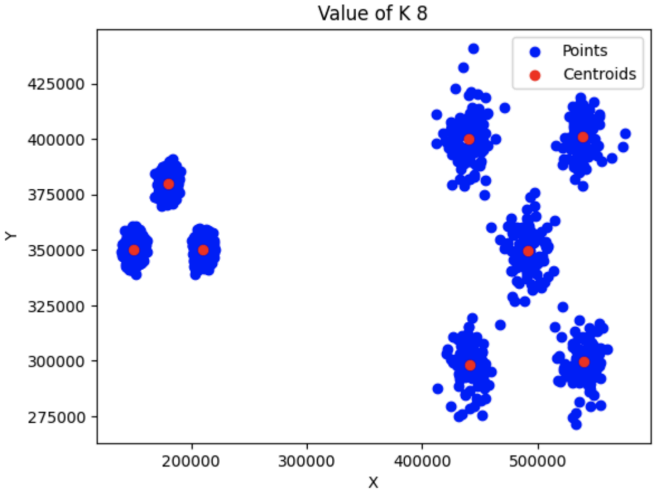
```
K = 5:
Avg WSSE: 3389790245740.807
Avg BSSE: 197249822873.5015
Avg SC: 0.5248172087194916

K = 6:
Avg WSSE: 1223056433655.749
Avg BSSE: 259888081578.61853
Avg SC: 0.9346752632223018

K = 7:
Avg WSSE: 467403158539.18555
Avg BSSE: 359309361347.8365
Avg SC: 0.9602680318560003

K = 8:
Avg WSSE: 214492062847.68344
Avg BSSE: 434210173757.1145
Avg SC: 0.9724023548273677

Best K: 8
```











Bisecting K-MEANS clustering using PYSPARK MLLIB Kmeans function on DS2, DS3

This section analyzes the findings of applying the Bisecting K-Means clustering algorithm using PySpark MLlib on dataset DS2. The objective was to partition the dataset into distinct clusters based on similarities between data points. We utilized the Bisecting K-Means algorithm from PySpark MLlib to perform clustering. The dataset DS2 was preprocessed and fed into the algorithm. We experimented with K values ranging from 4 to 8 and conducted each experiment three times to account for variability in cluster initialization.

The results of the experiments revealed interesting insights into the clustering performance at different values of K. As K increases, both the Within Set Sum of Squared Errors (WSSE) and Between Set Sum of Squared Errors (BSSE) tend to decrease. This is expected, as increasing the number of clusters allows the algorithm to better fit the data, resulting in smaller within-cluster variances. However, the rate of decrease diminishes as K grows larger, suggesting a point of diminishing returns.

In addition to WSSE and BSSE, we evaluated the quality of clustering using the Silhouette Coefficient (SC). Higher SC values indicate better-defined clusters. In our experiments, K=5 produced the highest average SC, indicating that it yielded the most distinct and well-separated clusters. However, it's important to note that higher SC values do not necessarily mean better clustering, as they may also indicate overfitting or the presence of outliers.

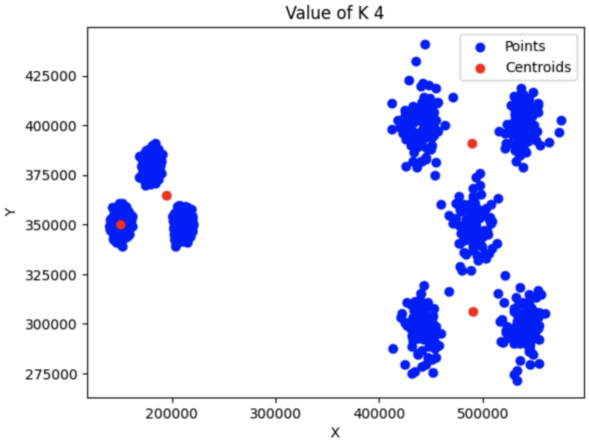
Based on the trade-off between WSSE, BSSE, and SC, we identified K=8 as the optimal number of clusters for the given dataset. K=8 produced the lowest average WSSE and BSSE among all tested values, indicating a good fit to the data. Additionally, it had a relatively high average SC, suggesting reasonably well-defined clusters. Therefore, K=8 is chosen as the optimal number of clusters for further analysis.

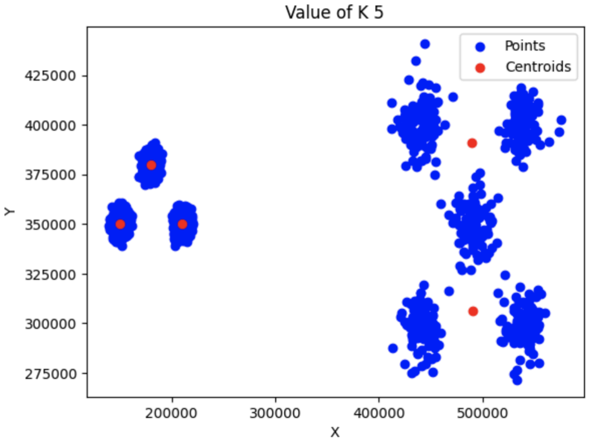
Results:

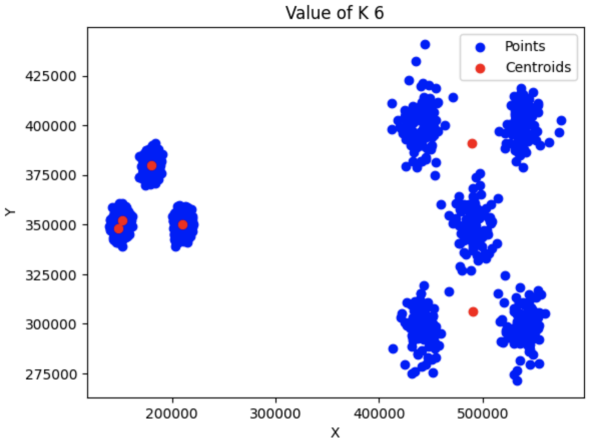
```
K = 4:
Avg WSSE: 3131880783349.231
Avg BSSE: 170725022484.73953
Avg SC: 0.7189355963485206

K = 5:
Avg WSSE: 1329600208960.2395
Avg BSSE: 171737896470.72723
Avg SC: 0.944259447127792

K = 6:
Avg WSSE: 1315002229668.3848
Avg BSSE: 174750869173.76517
Avg SC: 0.7933301534372686

K = 7:
Avg WSSE: 896170364009.0592
Avg BSSE: 267890792522.81213
Avg SC: 0.7952035170884736

K = 8:
Avg WSSE: 466530278713.2581
Avg BSSE: 349550008962.0465
```

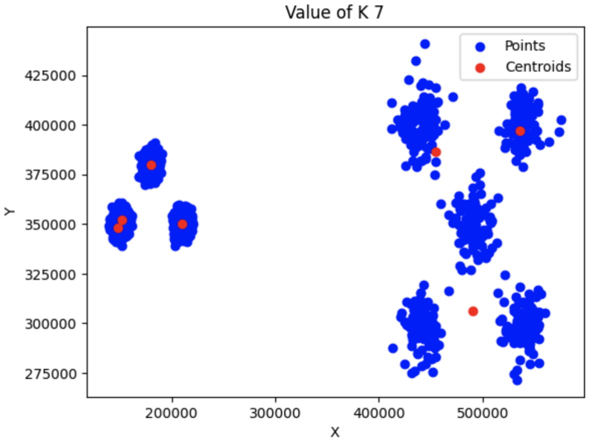
Avg SC: 0.8081755807545979

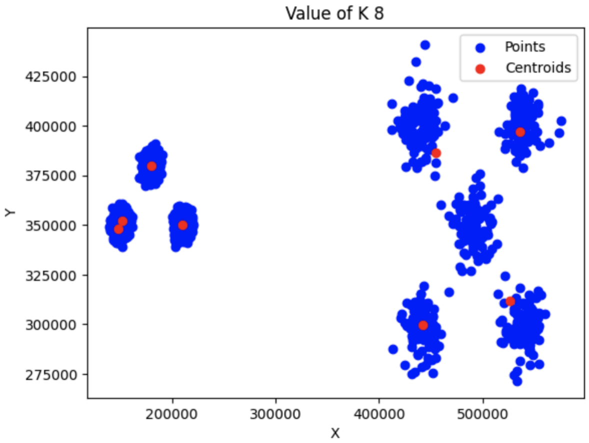
Best K: 8











Conclusion

The analysis of K-means clustering across datasets DS1 and DS2 highlights the importance of selecting an optimal number of clusters (K) to achieve meaningful results. In DS1, both standard and custom Bisecting K-means methods showed improved clustering performance with increasing K, as evidenced by decreasing WSSE. However, the fluctuating SC suggests varying levels of cluster cohesion and separation. For DS2, PySpark MLlib's KMeans function similarly demonstrated decreasing WSSE and BSSE with increasing K. The optimal K value of 8 yielded well-defined clusters, as indicated by a balance between WSSE, BSSE, and SC. Overall, the analysis emphasizes the need for careful consideration of K to ensure effective clustering and interpretation of results.