Colton Blackwell
301451278
Spring 2024

CMPT 459
Data Mining
Assignment 2: scRNA-seq Analysis

1) **Implement KMeans based on the provided code template.**

See kmeans.py and main.py included in the provided .zip file for details.

2) **Use random initialization to produce clusterings for k from 2 to 10 and plot the silhouette coefficient for all clusterings. What is the best k?**

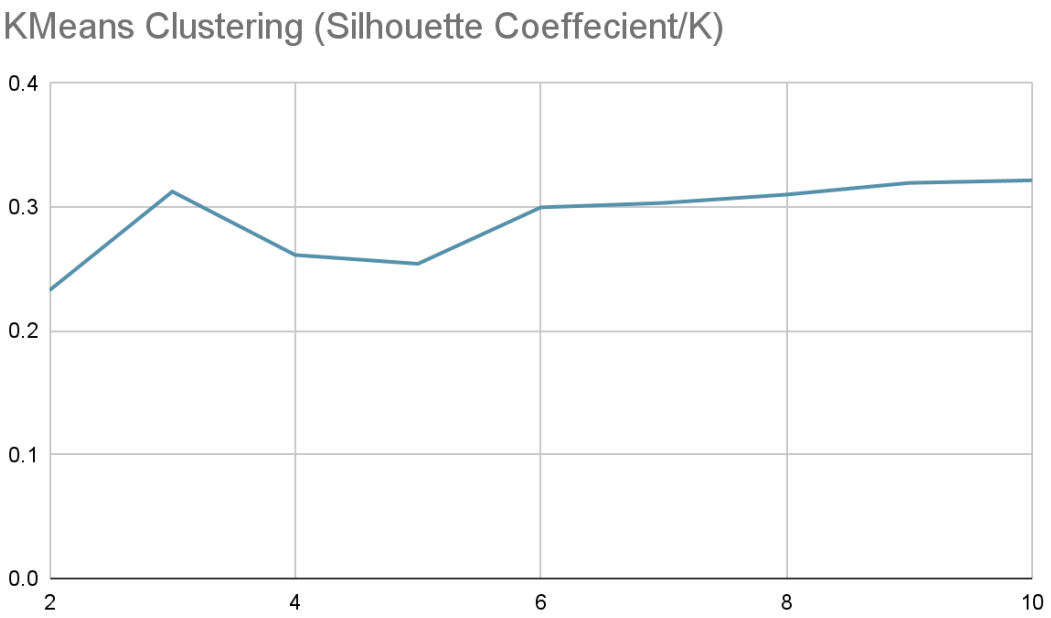
a. Random cluster algorithm implementation

The random cluster algorithm is implemented by randomly selecting K clusters from the dataset X within the range [0, number of samples). This approach guarantees that a data point is selected at most once, as replacement is prohibited during the selection process.

b. Relevant data

K clusters	*Silhouette Coefficient
2	0.2327812
3	0.3125352
4	0.2613056
5	0.2541977
6	0.2997631
7	0.3033568
8	0.3101016
9	0.3195487
10	0.3216201

*Given PCA dimensionality set to 100 / iterations set to 300



c. What is the best K?

The K that results in the highest Silhouette Coefficient (best K) is also the largest (K = 10). This pattern can be described by:

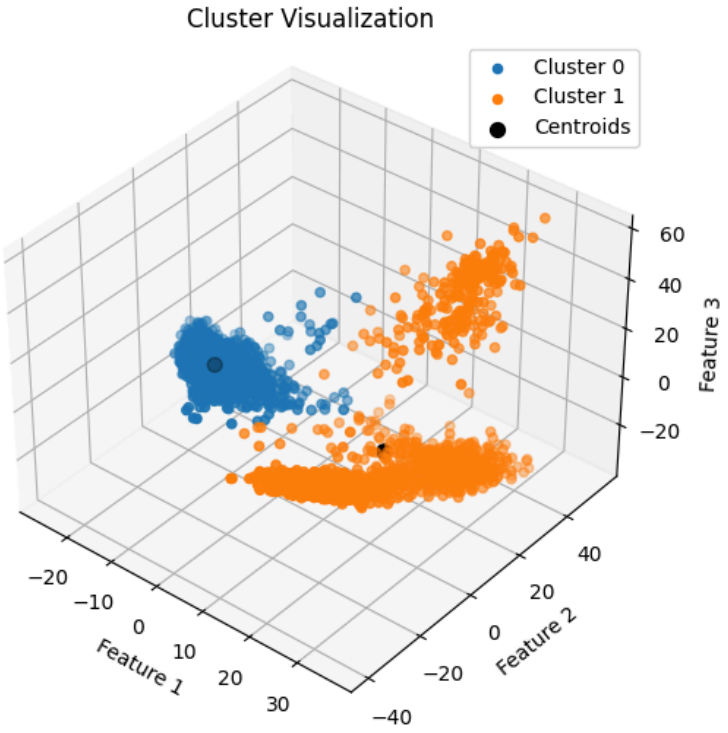
- When the data has more clusters (K), each data point is assigned to a smaller and more specific cluster. This leads to clusters that closely match the data points they contain, producing a higher silhouette coefficient.
- With a larger number of clusters, the centroids are more likely to be placed closer to dense data regions, which can lead to better separation between clusters.

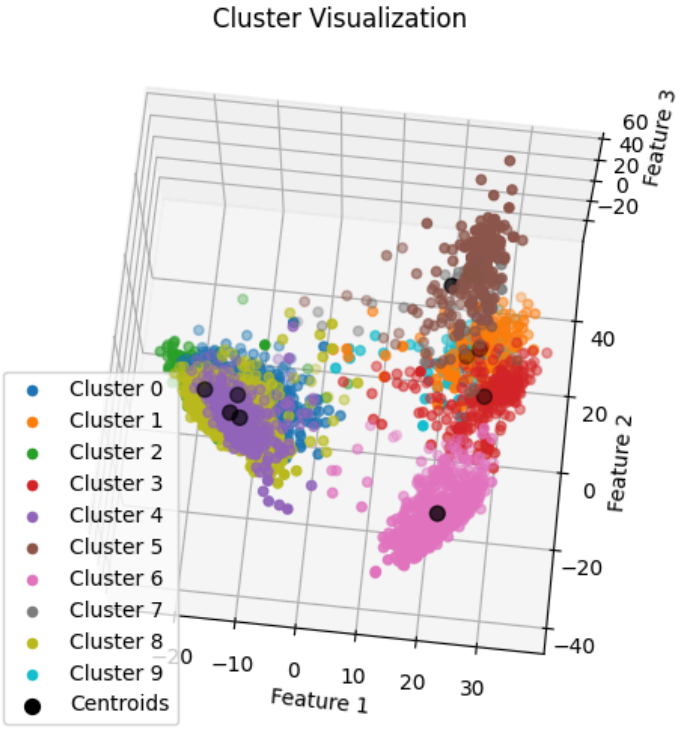
d. How did the dimensionality/# of iterations affect the Silhouette Coefficient/clustering algorithm?

As the dimensionality increased using the given PCA function, the Silhouette Coefficient appears to have decreased drastically. This may be due to the curse of dimensionality, in which in high dimensional spaces, data points tend to be evenly spaced from each other, leading to clusters that may not represent the underlying structure well.

Increasing the number of iterations in KMeans allows the algorithm more opportunities to converge to a better clustering solution. With more iterations, KMeans has a better chance of minimizing the within-cluster sum of squares and finding centroids more representative of the underlying clusters.

KMeans 3D Visuals (k = 2 vs k = 10)





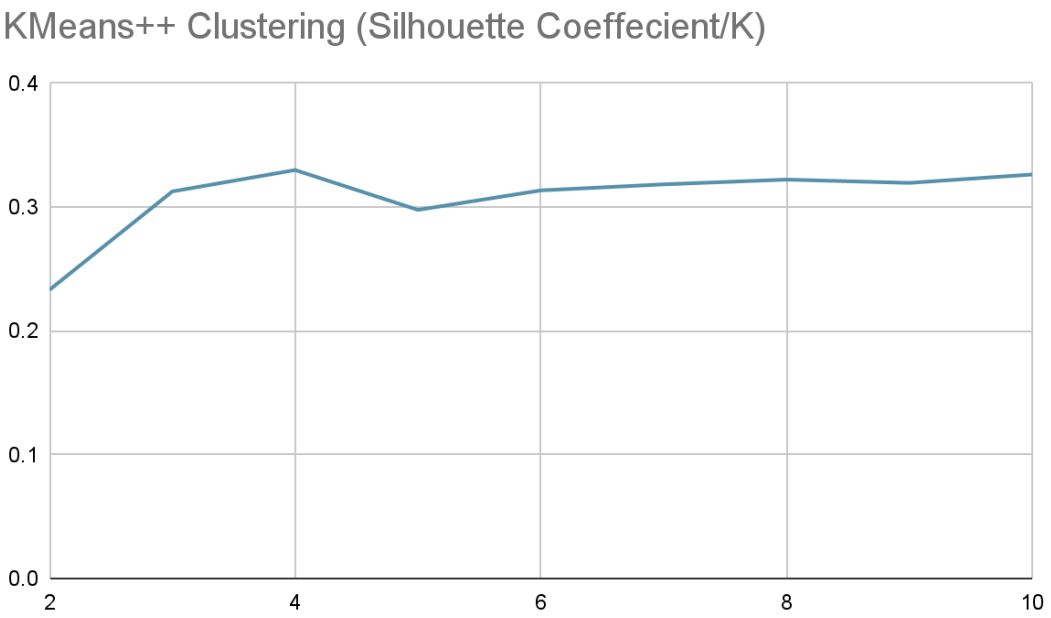
3) **Use kmeans++ initialization to produce clusterings for k from 2 to 10 and plot the silhouette coefficient for all clusterings. What is the best k?**

a. How the kmeans++ cluster scheme was implemented

The random cluster algorithm is implemented by computing the data point to the nearest centroid already chosen, finding the minimum distance for each point, and calculating the probabilities of selecting each data point as the next centroid. Lastly, we choose the next centroid by sampling from the data points with probabilities and assigning the selected data point as the next centroid

b. Relevant data

K clusters	***Silhouette Coeffecient**
2	0.2331649
3	0.3126540
4	0.3299333
5	0.2541977
6	0.2997631
7	0.3033568
8	0.3101016
9	0.3195487
10	0.3216201



c. What is the best K?

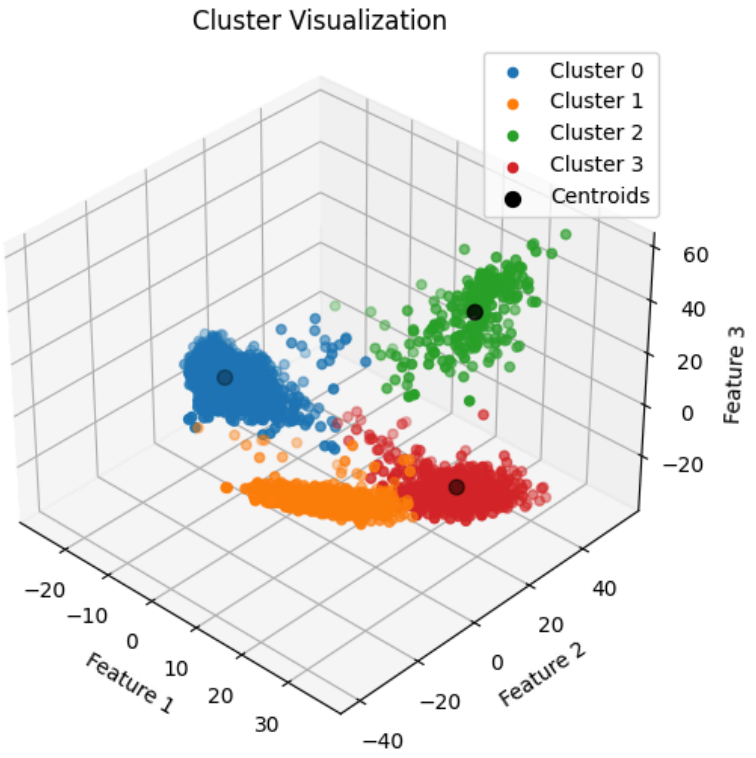
The optimal K within the range of [2,10] was found to be k = 4, based on the maximum Silhouette Coefficient. This value of k is considered more desirable due to its ability to strike a balance between underfitting and overfitting.

When there are too few clusters, it means the algorithm might oversimplify the structure of the data. In such cases, distinct patterns or groupings within the data might not be adequately captured. Consequently, the resulting clusters may lack granularity, leading to a loss of information and potential misrepresentation of the underlying data distribution. However, when there are too many clusters, the kmeans++ algorithm may overfit the data. This means it creates clusters that are highly specific to the training data. Overfitting can result in clusters that capture noise or outliers rather than meaningful patterns.

d. How did the dimensionality/# of iterations affect the Silhouette Coefficient/clustering algorithm?

Similar results are evident within the KMeans implementation; refer to point 2 d) for further details.

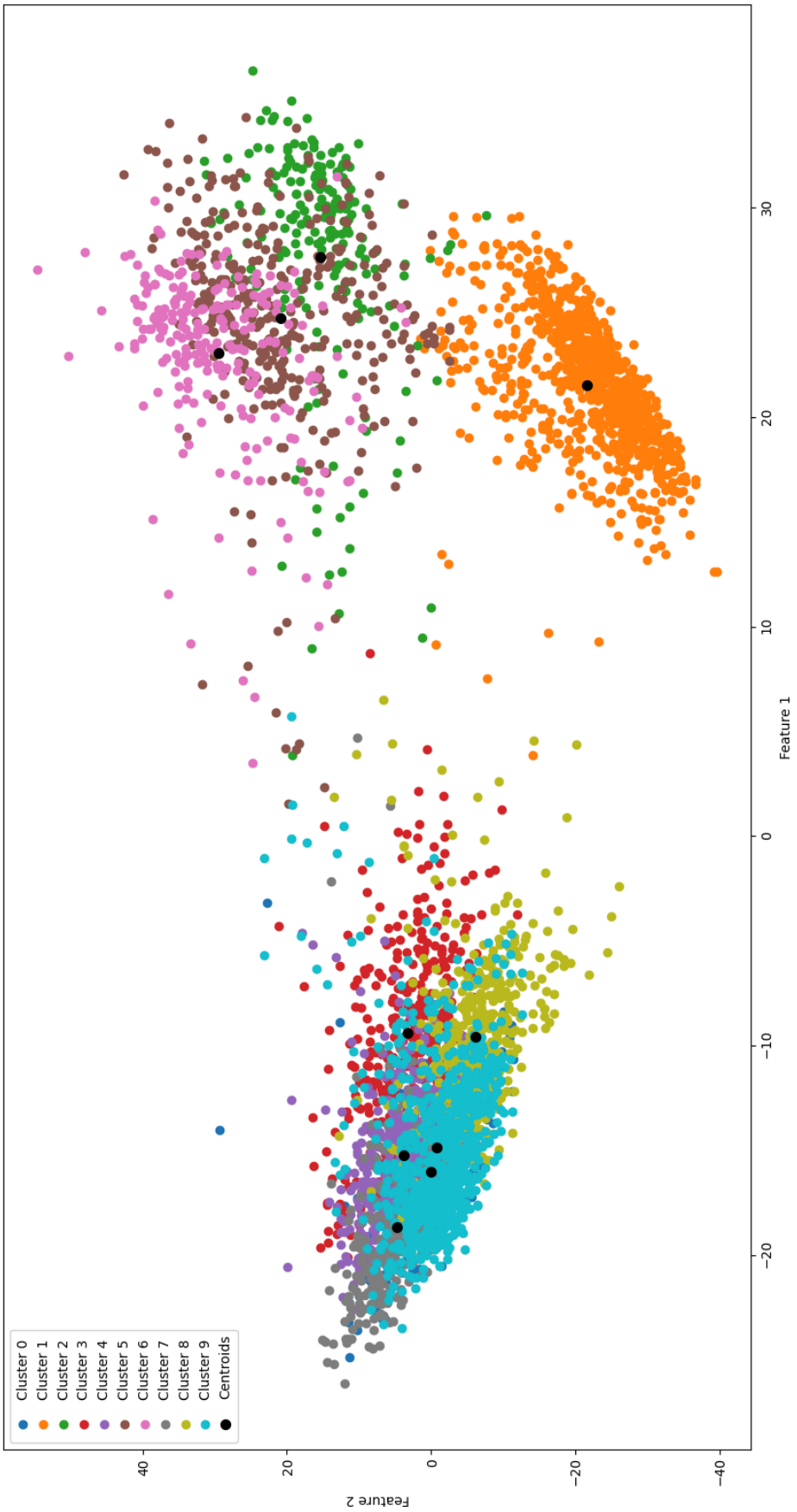
KMeans++ 3D Visual (k = 4)



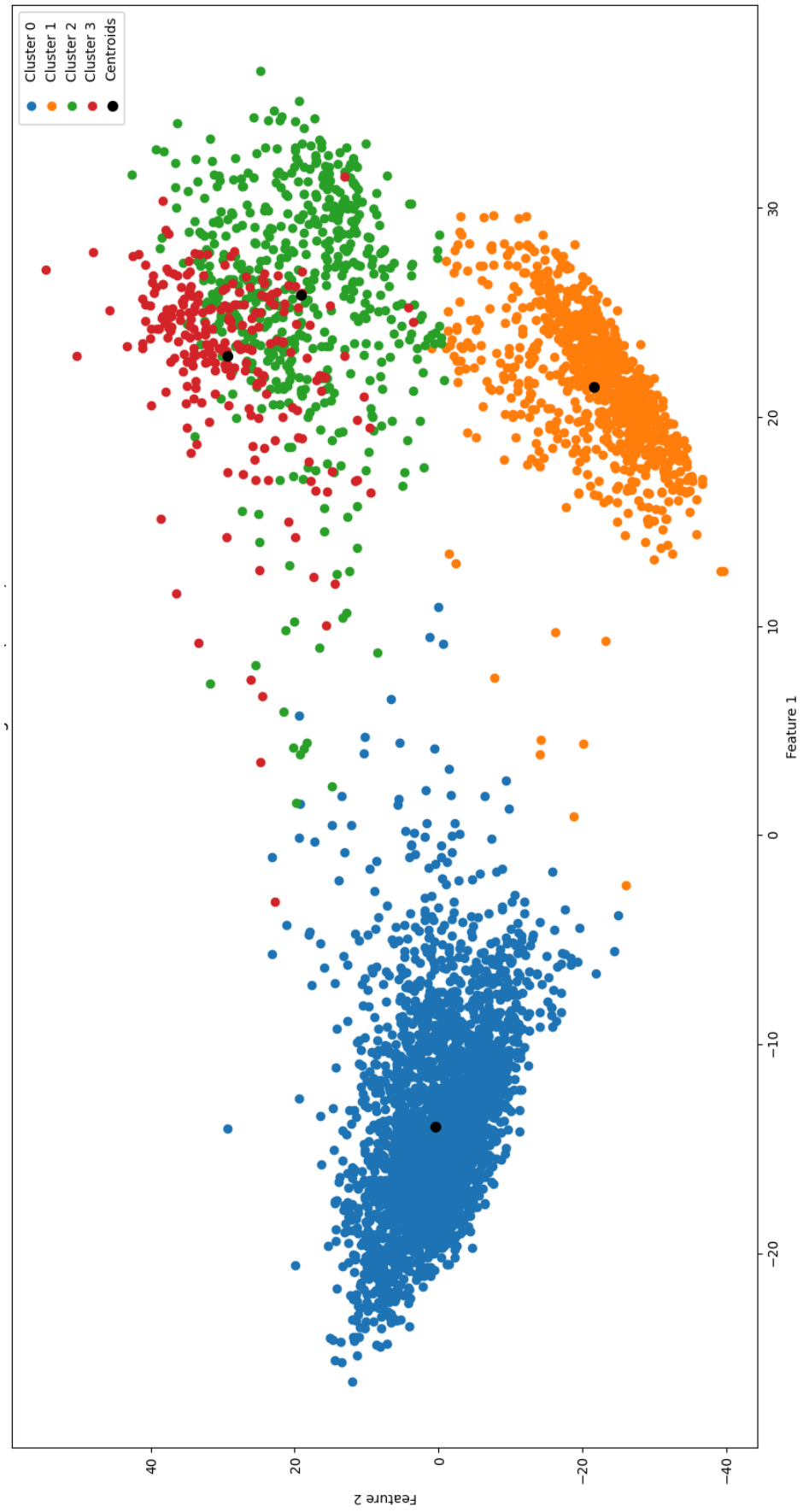
4) **Use a scatter plot to visualize the clusters with the best k from 2 and 3.**

See pages 6 and 7 for details.

KMeans 2D Visual - Best K (K = 10)



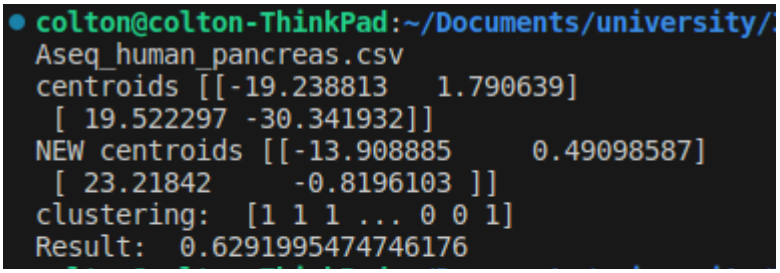
KMeans++ 2D Visual - Best K (K = 4)



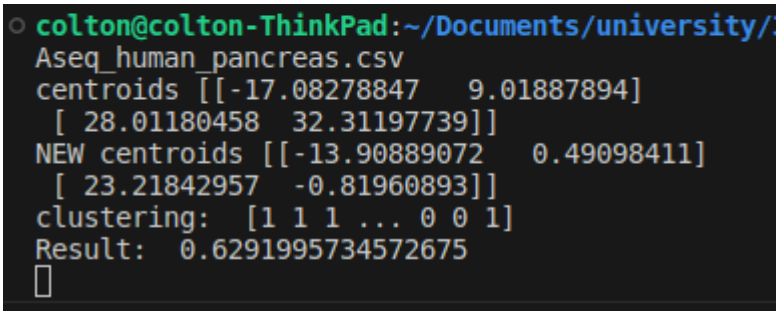
Additional Comparison Data

Terminology:

Centroids - initially chosen centroids
New centroids - Final chosen centroid after readjusting
Result - Silhouette Coefficient
Clustering - Array of class labels
Dimension - Set to 3rd Dimension

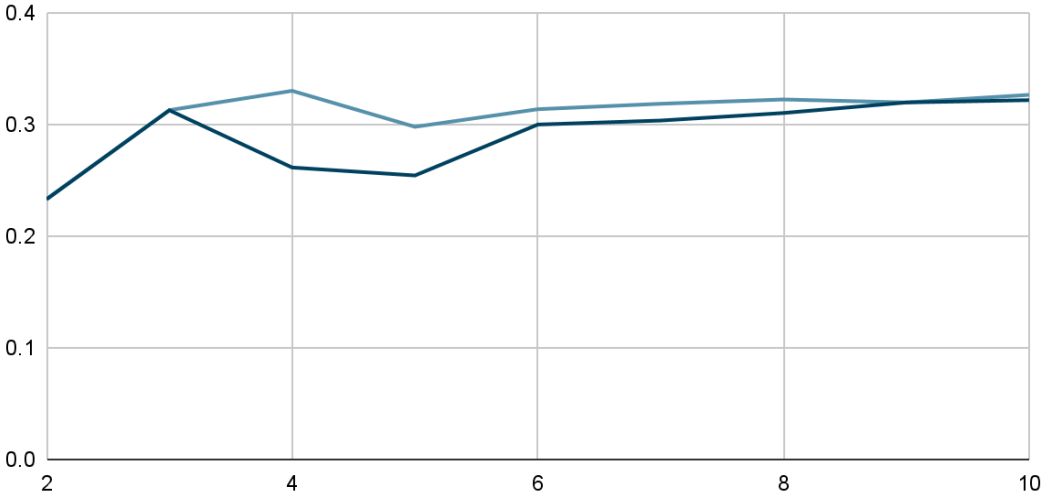


The silhouette coefficient result of KMeans



The silhouette coefficient result of KMeans++

The provided visuals illustrate that regardless of the centroid initialization method employed–be it KMeans or KMeans++–the resulting clusters appear virtually identical. This similarity arises from both algorithms iteratively adjusting centroids to optimize clustering. Therefore, any distinctions in cluster initialization have minimal impact on the overall outcome.



Light Blue = KMeans++
Dark Blue = KMeans

X-axis = K clusters
Y-axis = Silhouette Coefficient

When plotting KMeans++ and KMeans directly on the same line graph, KMeans++ typically carries a higher Y-axis value, as anticipated.