CMPT 459 Assignment 2

Name: Shan Ying (Felicity) Yang
Student Number: 301305759
Professor: Dr. Martin Ester
TAs: Arash Khoeini, Shuman Peng
Submission Date: March 4, 2022

Task 1

Please refer to the coding portion in kmeans.py for Task 1! I have added an "init" argument that specifies the initialization method for K-means/K-means++, by default init is random.
Example Command: `python main.py --n-clusters 0 --data "./Heart-counts.csv" -init "kmeans++"`

Task 2

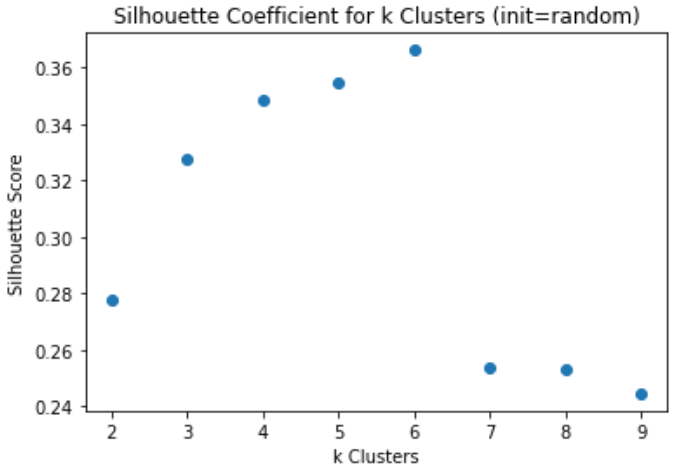
The best k clusters produced from 2 to 9 using K-means initialization is at 6, and its Silhouette-Coefficient is approximately 0.366271. Below is the output of different k's (n_clusters's) Silhouette scores ranging from 2 to 9 with K-means initialization.

Figure 1. Silhouette Score for Different k (K-means Initialization)

```
At k =  2 , Silhouette Score is:  0.2774414439592757
At k =  3 , Silhouette Score is:  0.32733962352782986
At k =  4 , Silhouette Score is:  0.3486917467108892
At k =  5 , Silhouette Score is:  0.3546035376041207
At k =  6 , Silhouette Score is:  0.3662712184707359
At k =  7 , Silhouette Score is:  0.2539509354917117
At k =  8 , Silhouette Score is:  0.25309870586075406
At k =  9 , Silhouette Score is:  0.24454368328690423
Best k is:  6 ; SC =  0.3662712184707359
```

The Silhouette-Coefficient's computation follows the Silhouette formula from the lecture note on Cluster Analysis: $s(o) = \frac{b(o)-a(o)}{\max\{a(o),b(o)\}}$. The variable $a(o)$ is the distance of object o to the nearest cluster's centroid (cluster representative). The variable $b(o)$ is the distance of object o to the representative of the second nearest cluster centroid. The final Silhouette-Coefficient is computed by averaging silhouette over all objects. For all $s(o)$ to be greater than and less than 0.5, the silhouette coefficients reflect that the cluster has a reasonable cluster structure for each corresponding k values. Below is a scatter plot of the K-means clustering's Silhouette-Coefficient for k from 2 to 9.

Figure 2. Scatterplot of Silhouette-Coefficient for k Clustering with K-means Initialization



From figure 2 we can examine that the Silhouette-Coefficient has an increasing trend as the k value increases before the maxima, where the maxima is the best k (k=6). After k=6, the Silhouette-Coefficient significantly drops over 0.11232, and continues to show a decreasing trend from k=7 onward.

Task 3

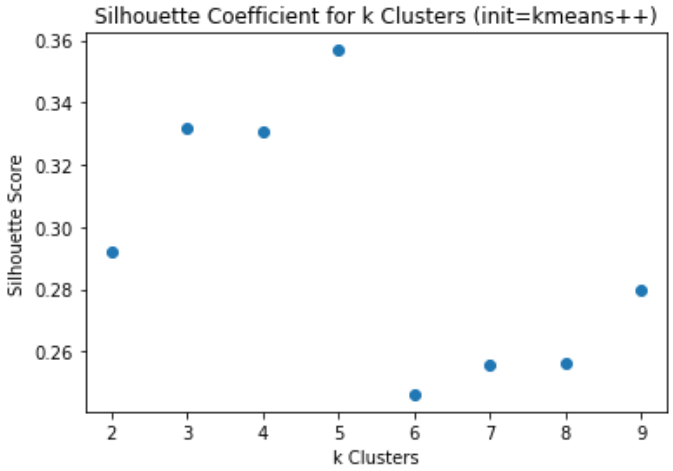
The best k clusters produced from 2 to 9 using K-means++ initialization is at 5, and its Silhouette-Coefficient is approximately 0.0.356859. Below is the output of different k's (n_clusters's) Silhouette scores ranging from 2 to 9 with K-means++ initialization.

Figure 3. Silhouette Score for Different k (K-means++ Initialization)

```
At k =  2 , Silhouette Score is:  0.29180299467674314
At k =  3 , Silhouette Score is:  0.3315271847245973
At k =  4 , Silhouette Score is:  0.33074003052942963
At k =  5 , Silhouette Score is:  0.35685901018969274
At k =  6 , Silhouette Score is:  0.24617051531125883
At k =  7 , Silhouette Score is:  0.2556807407739409
At k =  8 , Silhouette Score is:  0.25597657004185514
At k =  9 , Silhouette Score is:  0.2795941857037456
Best k is:  5 ; SC =  0.35685901018969274
```

Below is a scatter plot of the K-means++'s Silhouette-Coefficient for k from 2 to 9.

Figure 4. Scatterplot of Silhouette-Coefficient for k Clustering with K-means++ Initialization



From Figure 4 we can examine that the Silhouette-Coefficient has an increasing trend as the k value increases before the maxima, where the maxima is the best k (k=5). After k=5, the Silhouette-Coefficient significantly drops over 0.110688. Unlike K-means initialization, the Silhouette-Coefficient has an increasing trend from k=6 to k=9 after it drops to the minimum value at k=6.

The major difference between K-means and K-means++ is how each algorithm initializes the centroids. While K-means picks the initial centroids randomly, K-means++ randomly chooses the first centroid and chooses the subsequent centroids according to the probability proportional to the Euclidean distance of each data point. Table 1 below compares the Silhouette-Coefficient of K-means and K-means++.

Table 1. Comparison of Silhouette-Coefficient of K-means and K-means++

	K-means	K-means++	Differences
$k = 2$	0.2774414439592757	0.29810299467674314	0.020661551
$k = 3$	0.32733962352782986	0.3315271847245973	0.004187561
$k = 4$	0.3486917467108892	0.33074003052942963	−0.01795172
$k = 5$	0.35446035376041207	0.35685901018969274	0.002398656
$k = 6$	0.3662712184707359	0.24617051531125883	−0.1201007
$k = 7$	0.2539509354917117	0.2556807407739409	0.001729805
$k = 8$	0.25309870586075406	0.25597657004185514	0.002877864
$k = 9$	0.24454368328690423	0.2795941857037456	0.035050502

Figure 5. Silhouette Coefficient of K-means and K-means++ with different k values

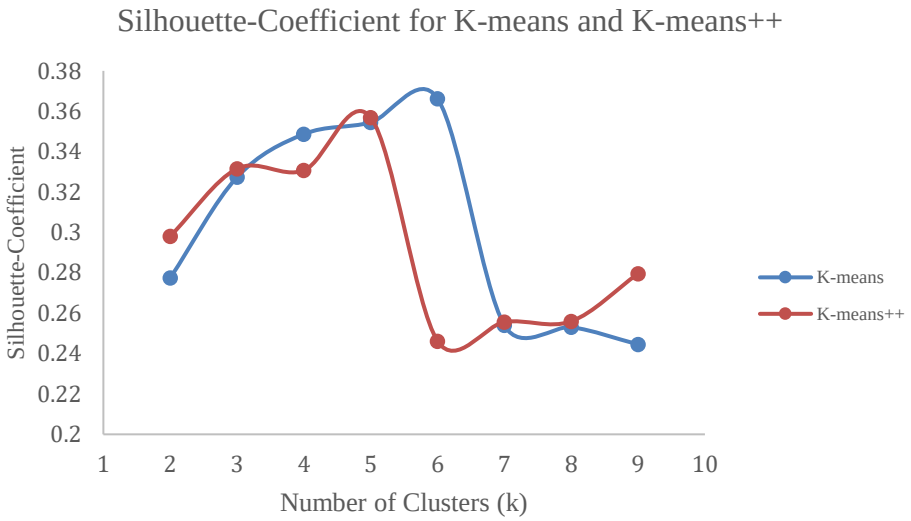


Figure 5 combines Figure 2 and Figure 4, and we can see that for most of the time, K-means++ performs better than K-means. The overall trendlines for both K-means and K-means++ are similar, where they both reach a maximum Silhouette-Coefficient midway and drops to a low Silhouette-Coefficient. The only k values that K-means outperforms K-means++ is at k=4 and k=6, this could be due to inconsistent predicting that my K-means++ algorithm has. However, after K-means++'s Silhouette-Coefficient drops to its minimum value, the score increases again for k=7 and onwards. This may imply that for larger number of clusters (larger k values), K-means++ tends to stably outperforms K-means. Also, the model is fitted on a dataset with 100 column features, which may also contribute in affect the performance of my K-means and K-means++ algorithms. In conclusion, K-means++ generally should perform better than K-means, and at higher k values K-means tend to perform worser than K-means++.

Task 4

Command to compute Task 4: `python main.py --n-clusters 0 --data "./Heart-counts.csv"`
I have set n_clusters to 0 to be performing task 4. Using other n_clusters values greater than 0 will compute the K-means cluster with the specified k, and cluster plot with PCA0 and PCA1 will be plotted.
Figures 6 and 7 show the visualization of the clusters with the best k=6 from task 2. PCA# refers to the column feature of the heart cell dataset after reducing dimensionality to 100 with PCA. PCA0 refers to the first column of the dataset, and PCA1 referst to the second column of the dataset. The clusters are also color-coded with 6 different colors. However, because PCA0 and PCA1 is harder to see the sixth corlor, I included Figure 7 to show that K-means plotted 6 clusters.

Figure 6. Cluster Visualization with PCA0 and PCA1

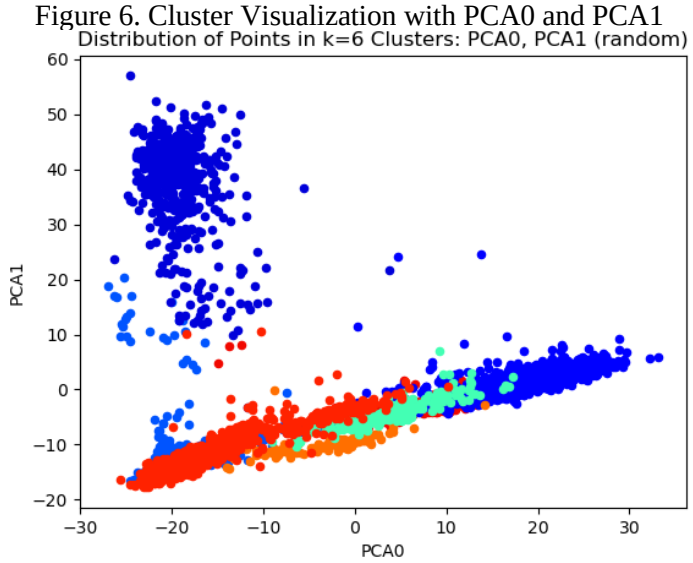


Figure 7. Cluster Visualization with PCA1 and PCA2