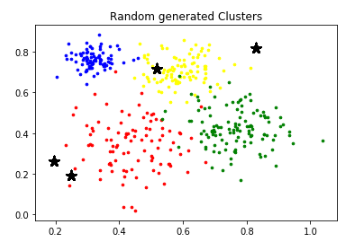
**Part1 Given Dataset using K means:**

Since the dataset has 4 columns with ID, X1, X2, TrueCluster, I extract X1, X2 as the dataset to perform K means Algorithm. First of all, I use the scatter plot with the trueCluster as the color to see the original clusters. Therefore, I need to find the true centroids first, which is the mean of all the data points in each cluster.

**Initialization of random centroids:**

Then I initialize 4 random generated centroids and multiplies the standard dervation and adds the mean of the dataset.



**Update of centroids:**

Since we need to calculate the distance between each data points to the centroids, Euclidean distance is a better option. By using Euclidean distance, I calculate the distance between each data point and the centroids. In order to do that, I create a function that returns the distance matrix with the 400\*4 size of all data points to each centroid. After that, for each data point, I pick the minimum Euclidean distance and assign the new centroids by calculating the mean of the sum of the minimum distance of all data points.

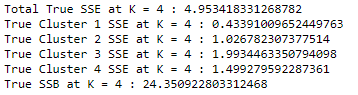
**Termination condition:**

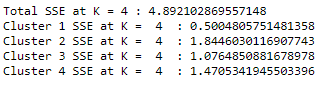
The termination condition is when the new assigned centroids is the same as the old centroids. Therefore, I add a while loop checking whether the new centroids are the same as the old centroids. Furthermore, I check the centroids change at the end of the loop so that it will terminate if the new centroids are the same as the old centroids.

**Validation measure:**

In order to validate the result of the new clusters, I implement SSE and SSB internal measures. By calculating the SSE, I create four variables with the summation of all the distance in each cluster. By calculating the SSB, I simply just do a subtraction of the SSE when only one cluster is assigned and SSE when four clusters are assigned.

**SSE generated by K means algorithm**

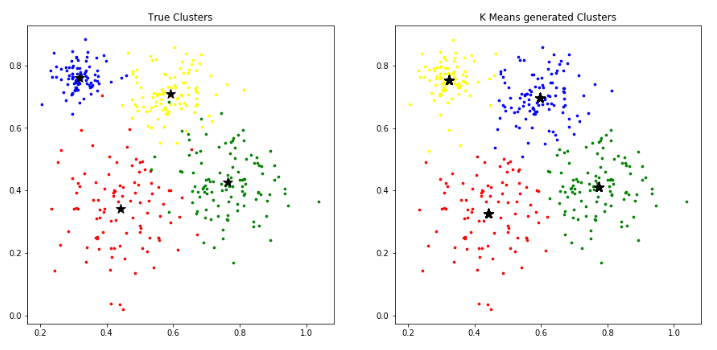






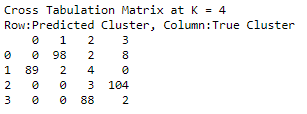
**Comparison**:

Since the true clusters are given, I plot the true centroids and color each cluster differently. In addition, I plot the centroids with K means algorithm and color each cluster differently. Based on the comparison of the two graphs, it is clear that in the True Cluster, there are a few red data points in the green cluster, and there is a red data point near the blue cluster, but in K Means generated Clusters, these unique points are assigned into their nearest clusters.



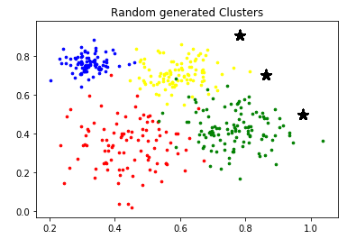
**Cross Tabulation Matrix:**

Since the both of true clusters and assigned clusters have 4 clusters. By using cross-tabulation matrix, we can clearly see the comparison between predicted result and the actual clusters. The cross-tabulation matrix below shows that the predicted result has high similarity with the actual result. The order of the cluster does not matter since there is no specific order to compare with the true cluster. By looking at the matrix from the column and row, each of the actual clusters is very intense, since each column has a relatively larger value than the rest. Furthermore, the small values in each column are classified to other clusters by K means algorithm. Since these small values are very small, it means the majority of data points are classified into the correct clusters. Therefore, this dataset should have pretty clear cluster separations.

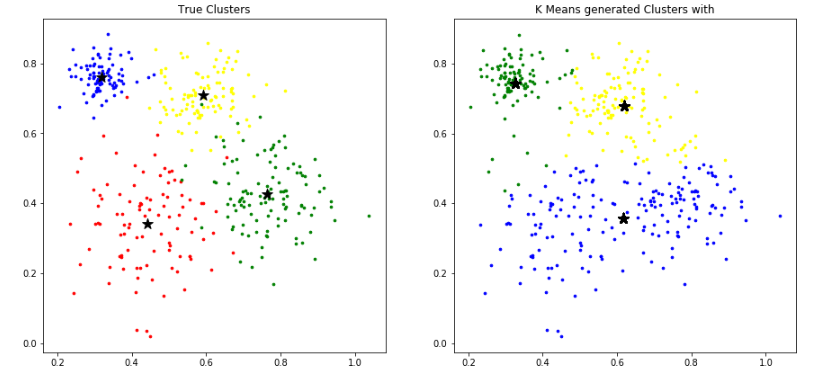


**K means with K = 3:**

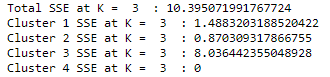
**Random generated centroids with true cluster coloring.**



**New centroids run by K means algorithm.**

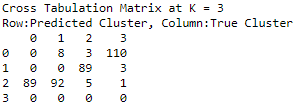


**SSE & SSB with K = 3:**





**Cross Tabulation Matrix at K = 3:**



**Analysis:**

Apparently, if K is equal to 3 instead of 4, the total SSE increase from 4.48921 to 10.39507. It means that the total distance between all data points and centroids within in one cluster for K = 3 is relatively larger than the total distance between all data points and centroids within in one cluster for K = 4. Furthermore, the SSB at K =3 is less than the SSB at K = 4. It means the distance between each cluster is closer when K is equal to 3 than the distance between each cluster when K is equal to 4. Furthermore, the SSE for each of the clusters increases, which means the data points in each cluster have longer distance compared with K = 4. The SSE for cluster 3 at K =3 has a large value compared with other clusters. Which means this cluster has pretty sparse data points. In addition, the cross-tabulation matrix for both clustering results is also changed a lot. Compared with the clustering with K =4, only a small portion of data is classified into other clusters based on the true cluster classification. However, we can see clearly that the third row of the cross-tabulation matrix at K = 3 has two relatively similar number of data points, which means this is a combination of the majority of data points of two true clusters. In conclusion, changing the number of clusters from 4 to 3 is worse based on the validation measures.

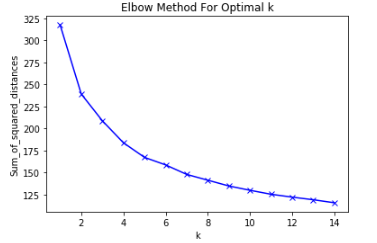
**Part 2 Wine Dataset:**

In this Wine Dataset, I decide to use the K means algorithm, Agglomerative Hierarchical algorithm, and DBSCAN algorithm. First, I drop attribute “Quality” because this is the true cluster and it shouldn’t be used in the clustering algorithm. Since each attribute of the Wine Dataset has large variance, I first normalize the whole dataset from 0 to 1. Then I proceed with each of the algorithms.

**K means:**

Since K means is a distance-based algorithm, normalization for the whole dataset is very necessary. In order to find the optimal K values for this Wine dataset, I calculate the sum of squared distance for K from 1 to 15. It turns out K = 6 is the optimal K value by considering the overfitting and underfitting.

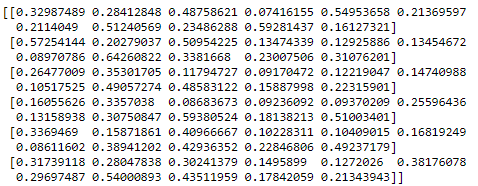
**Choice of K value:**



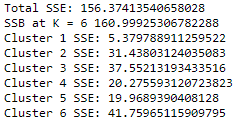
By running the K means algorithm from sklearn.cluster package with K = 6, I have the list of centroids. In order to understand whether the list of centroids has the best performance, I calculate the Euclidean distance between each data points and the centroids, since K means is a distance-based algorithm. Then I calculate the SSE, SSB, purity, and cross-tabulation matrix to make further analysis.

**Statistics:**

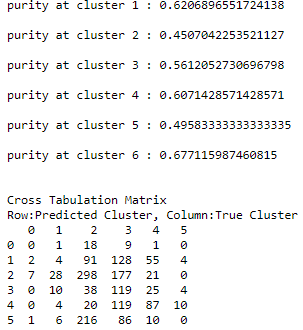
**Centroids when K = 6:**



**Here is the total SSE, SSB, and SSE for each cluster using K means algorithm.**



**Here is the purity of each cluster and the Cross-Tabulation Matrix using K means algorithm.**



**Analysis:**

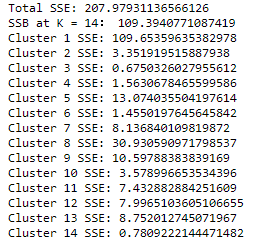
The performance of K means algorithm shown above can be illustrated as follows. Since we have 11 attributes, we cannot possibly have a visual image of the clustering image. Therefore, I start to analysis the statistic data to understand the performance of K means algorithm. First, I choose K = 6 from the elbow graph, which is the optimal K for this Wine dataset. Because in the elbow graph, the curve starts going smooth after K = 6, which means it will have an overfitting problem when K is greater than 6. Furthermore, When K is less than 6, the curve drops significantly, which means it will have and underfitting problem. Therefore, K = 6 is a relatively optimal choice for this dataset. Furthermore, from the internal measure of cluster validity perspective, the SSE can represent the distance between data points in each cluster to the centroids. Except for the first cluster with SSE = 5.37, the rest of clusters has relatively large SSE. That means the majority of clusters has sparse data points distribution. The SSB is also very large, which means each of the clusters has relatively clear separations. From the external measure of cluster validity perspective by including the true cluster attribute “Quality”, each of the purity of the clusters is from 45% to 67%. Because only one cluster has 67% of the purity and the rest of cluster purity is pretty average. Furthermore, the Cross-Tabulation Matrix also can give me some insights into the performance of K means algorithm. The matrix shows that the predicted clusters are totally different from the true cluster. Because each cluster doesn’t have a relatively large value that can represent the cluster. Therefore, the performance of K means algorithm for this Wine dataset is pretty average.

**DBSCAN**:

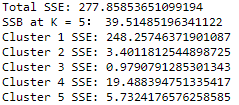
The value for the epsilon I choose is 0.2 and the min\_samples is 5 after many testing from the results of SSE and SSB. Since the whole dataset is scaled from 0 to 1, the epsilon shouldn’t be greater than 1. 0.2 is very reasonable since each of cluster won’t have too many data points. The value for min\_samples is also reasonable because if the value for min\_samples is too low, it will include many noises. Even though when epsilon is 0.2 and min\_sampoles is 3, the SSE is 207.97 and that is less than the SSE when epsilon is 0.2 and min\_sample is 5. However, it has 14 clusters, but only one cluster contains the most data points. Therefore, I think it is pretty acceptable that the value for epsilon is 0.2 and the value of min\_samples is 5 since it has fewer clusters. It turns out I have 6 clusters for this Wine dataset.

**Testing the epsilon and min samples:**

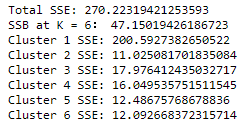
When Epsilon = 0.2, min\_samples = 3, the DBSCAN algorithm gives 14 clusters.



When Epsilon = 0.3, min\_samples = 3, the DBSCAN algorithm gives 5 clusters.



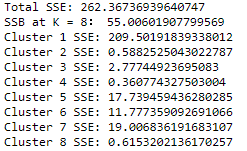
When Epsilon = 0.2, min\_samples = 5, the DBSCAN algorithm gives 6 clusters.



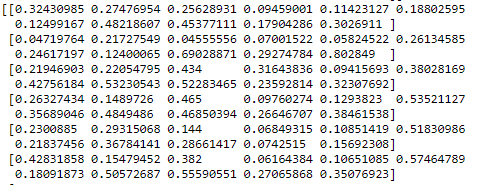
When Epsilon = 0.2, min\_samples = 5, the DBSCAN algorithm gives 1 cluster.



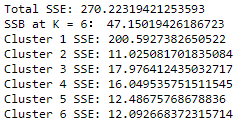
When Epsilon = 0.25, min\_samples = 3, the DBSCAN algorithm gives 1 cluster.



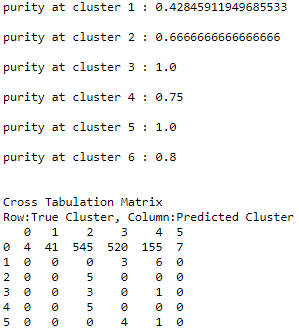
**In order to calculate the internal measures of cluster validity for epsilon = 0.2 and min samples = 5, I calculate the centroids of each clusters by taking the mean of the data points of each cluster.**



**Here is the total SSE, SSB, and SSE for each cluster using K means algorithm.**



**Here is the purity of each cluster and the Cross-Tabulation Matrix using K means algorithm.**

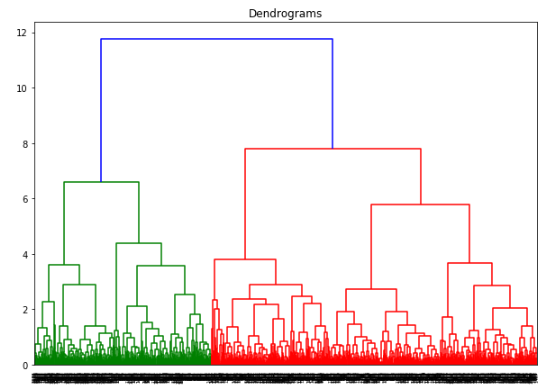


**Analysis:**

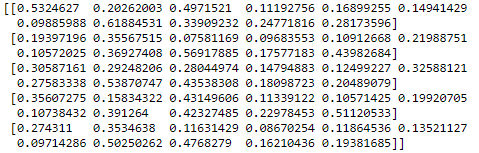
Based on the choice of the epsilon and min\_samples, 6 clusters are created. From the internal measure of cluster validity perspective, the SSE can represent the distance between data points in each cluster to the centroids. The SSE of the first cluster is 200.59, which is pretty large, which means the data points in the first cluster are pretty sparse. However, in other clusters, the SSE is significantly smaller than the first clusters. Furthermore, the SSB is also very small compared with the total SSE, which means each of the clusters has no relatively clear separations. From the external measure of cluster validity perspective by including the true cluster attribute “Quality”, except for the first cluster, the purity of rest of the cluster is pretty large, but there are only a few points in the rest of clusters. Therefore, high purity cannot represent the general performance. However, in the first cluster that contains the most data points, purity is only 42.8%. Overall, the performance of DBSCAN isn’t going well. Furthermore, the Cross-Tabulation Matrix also shows that the majority of data points is in cluster 1 and there are only a few data points in the rest clusters. Therefore, DBSCAN algorithm doesn’t classify the data points accurately. I can then conclude that the performance of the DBCAN algorithm is very low.

**Agglomerative Hierarchical:**

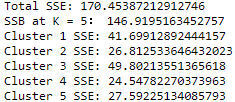
In order to find the optimal clusters numbers for Agglomerative Hierarchical algorithm, I create the Dendrogram graph. Since the horizontal line is the threshold, it defines the minimum distance to form a cluster. The optimal number of clusters is the number of lines that are the longest vertical line without horizontal line cross through. Therefore, in order to find the optimal number of clusters for the wine dataset of 1599 data points, I choose 5 clusters.



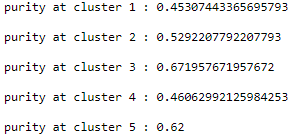
**Here is the graph of centroids of the 5 clusters:**

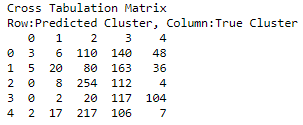


**Here is the total SSE, SSB, and SSE for each cluster using K means algorithm.**



**Here is the purity of each cluster and the Cross-Tabulation Matrix using K means algorithm.**





**Analysis:**

From the dendrogram graph, I pick only 5 number of clusters. Therefore, it won’t have overfitting or underfitting issues based on my choosing method illustrated above. Furthermore, from the internal measure of cluster validity perspective, the SSE can represent the distance between data points in each cluster to the centroids. The SSE of each cluster is pretty large, which means the data points in each cluster are pretty sparse. Furthermore, the SSB is also very large, which means each of the clusters has relatively clear separations. From the external measure of cluster validity perspective by including the true cluster attribute “Quality”, each of the purity of the clusters is relatively low except for the cluster 3 with 67% of purity. Furthermore, the Cross-Tabulation Matrix also can give me some insights into the performance of Agglomerative Hierarchical algorithm. The matrix shows that the predicted clusters are totally different from the true cluster. Because each cluster doesn’t have a relatively large value that can represent the cluster. this matrix shows the Agglomerative Hierarchical algorithm doesn’t classify the data points accurately. Therefore, the performance of Agglomerative Hierarchical algorithm for this Wine dataset is also pretty average.

**In conclusion:**

Based on the analysis and overall performances among the three clustering algorithms, I think K means algorithm has better performance for the Wine dataset. Since I have proceeded the internal and the external measures of cluster validity, K means algorithm has the less SSE and higher SSB, which means each data points are closer to the centroids and each cluster has more clear separation compared with the other two clustering algorithms. Furthermore, the overall purity of each cluster in the K means algorithms are relatively higher than the overall purity of each cluster in the other two clustering algorithms.