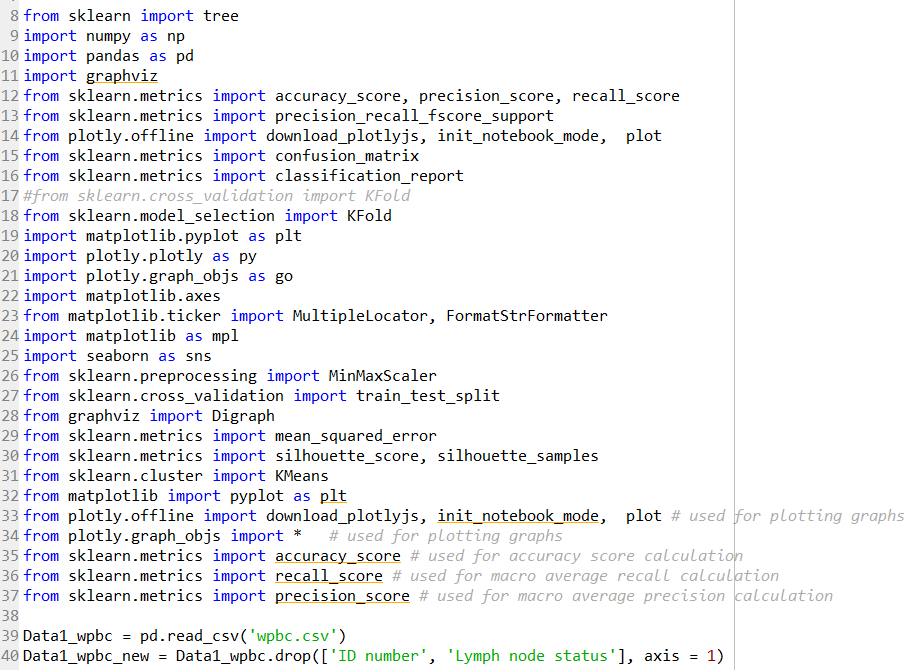
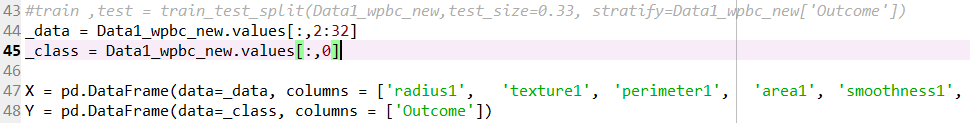
**K-Means**

**Considering the BCP dataset and its class variable with values “R” (Recurrence Occurred) and “N” (No Recurrence Occurred so far). Ignoring the attribute that gives the number of years after which recurrence occurred or the number of years for which the patient has been free of recurrence. There are thirty other attribute values given as features measured for every patient. Using only these thirty attributes.**

Import necessary packages. Import the wpbc data.

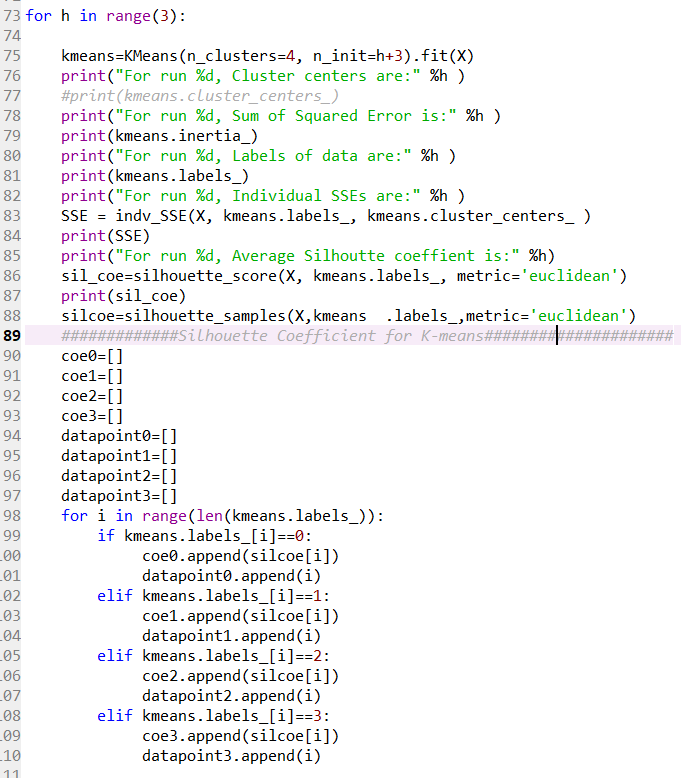


Separate the data into attributes (30) and target column.

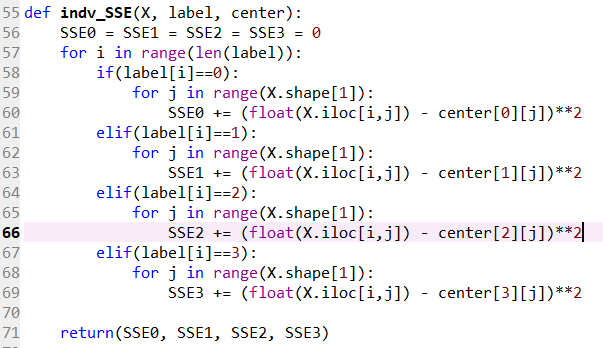


**Running k-means algorithm with this dataset for k=4. Repeating it three different times and for each run displaying the cluster centers and the SSE values for each cluster and also the total SSE value for the clustering.**

Running a k-means algorithm with cluster parameter set as 4. Set number of times the k-means algorithm will be run with different centroid seeds as iterator plus 3. Run the k-means for a total of 3 iterations to get the best results of the runs.



Calculating the individual SSE by using Euclidian distance measure for each cluster.



**OUPUT:**

The total SSE value for the clustering comes out to be the following.







The individual SSE values for each cluster are:

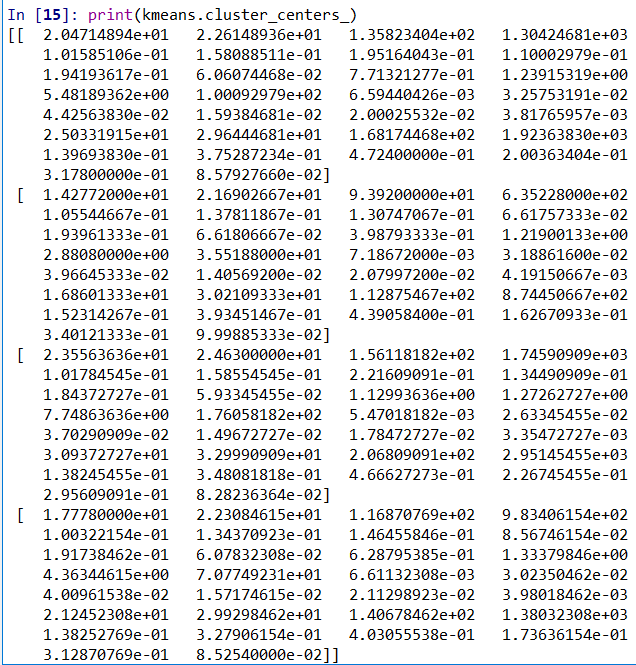
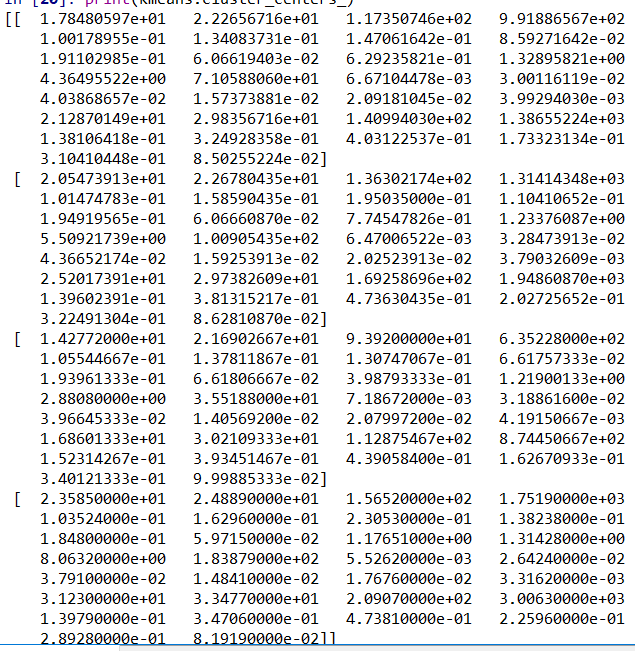




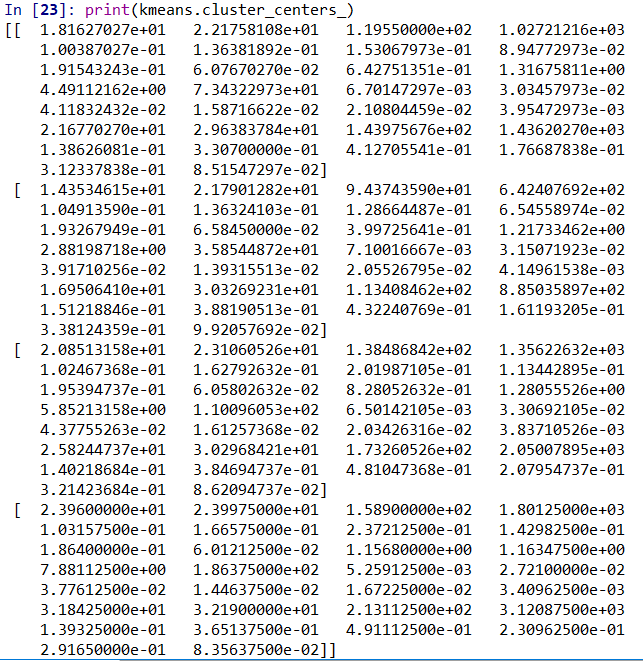


The cluster center coordinates having 30 values per coordinate value for each cluster are:

Run 0 Run 1

Run 2:

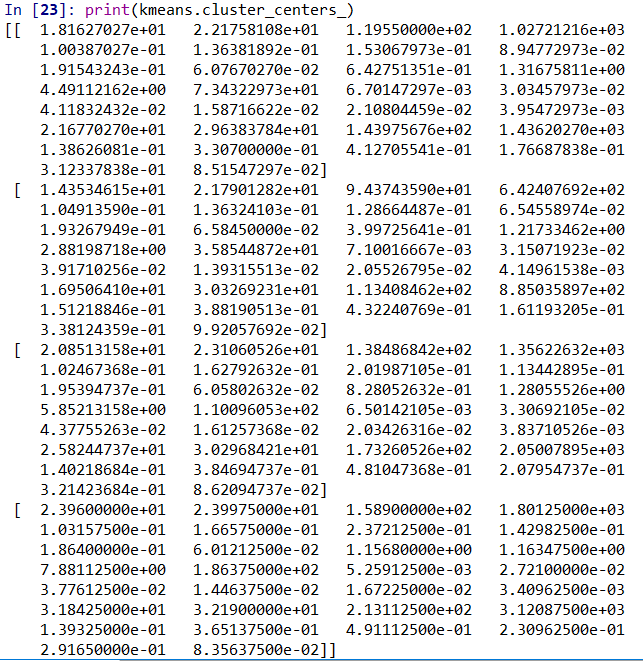


**Selecting the best of the above three clustering:**

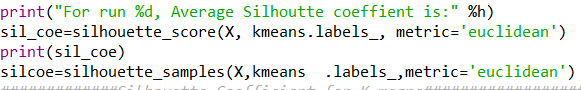
After running k-means for 3 iterations, the best minimum SSE values came in run 2. The criterion used here for choosing the best clustering is the total SSE value. SSE is the measure that K-means tries to minimize when the number of clusters is given. The idea is that we want a small SSE, but that the SSE tends to decrease toward 0 as we increase k. Thus, keeping k a constant at 4, iterating 3 times gives different clustering with different total SSE.





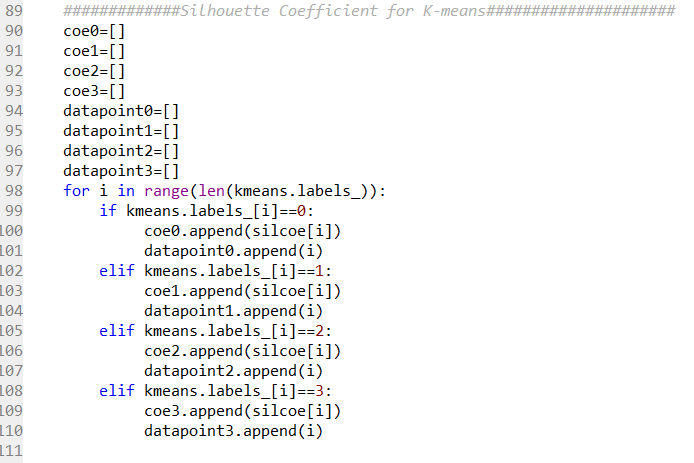


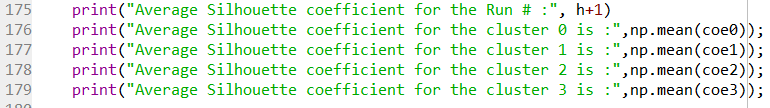
**For the best candidate clustering chosen, plotting the Silhouette coefficient for the clustering. Computing the average Silhouette coefficient for each cluster of the chosen clustering.**



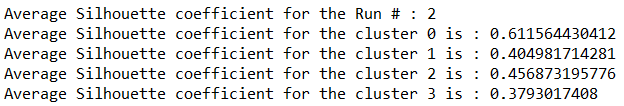
Average Silhouette coefficient for the clustering in run 2 is



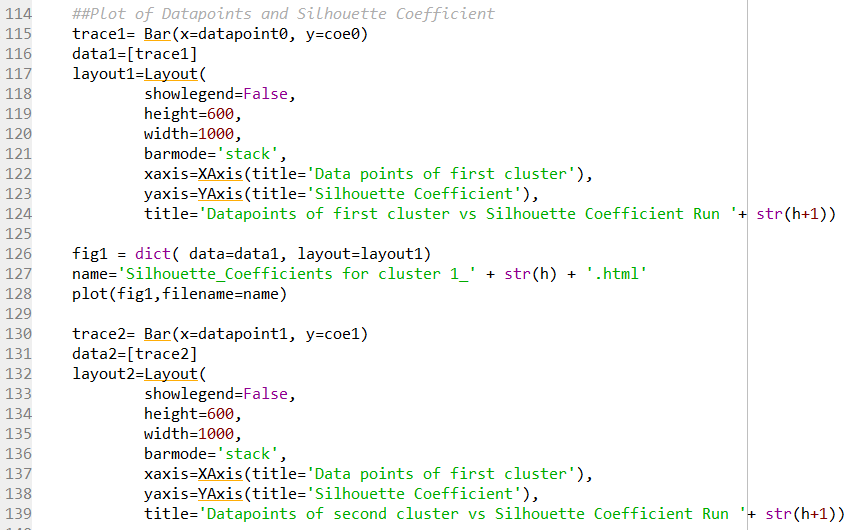


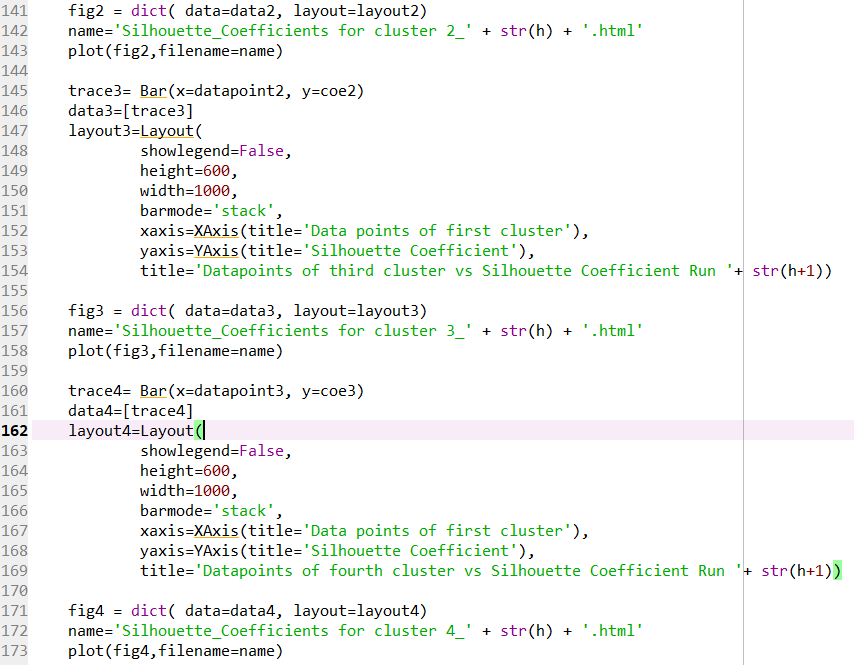


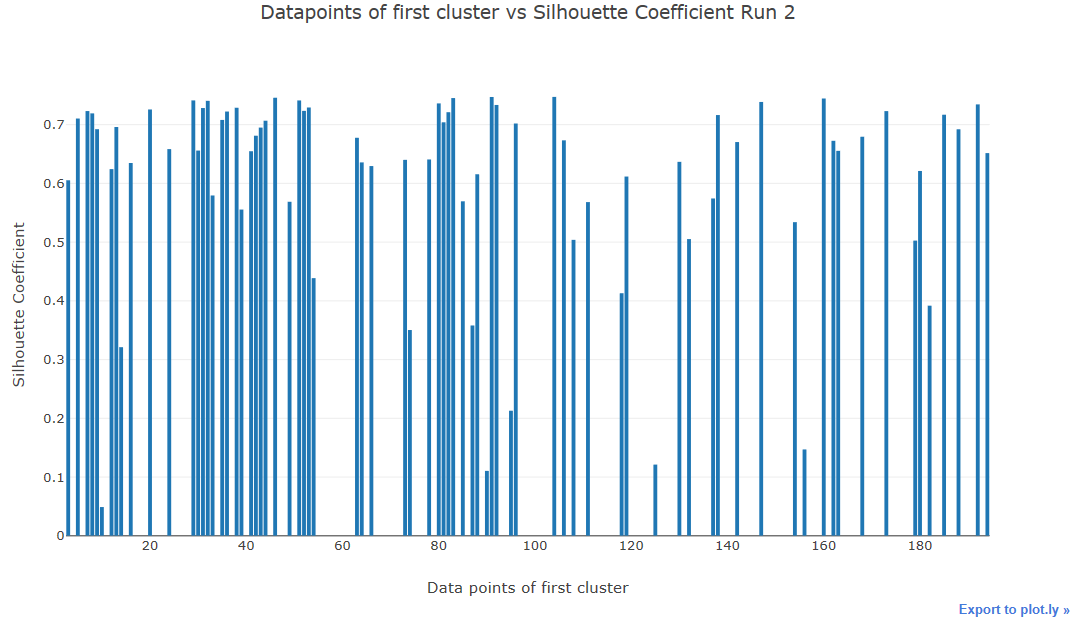
The average silhouette coefficient for each cluster is

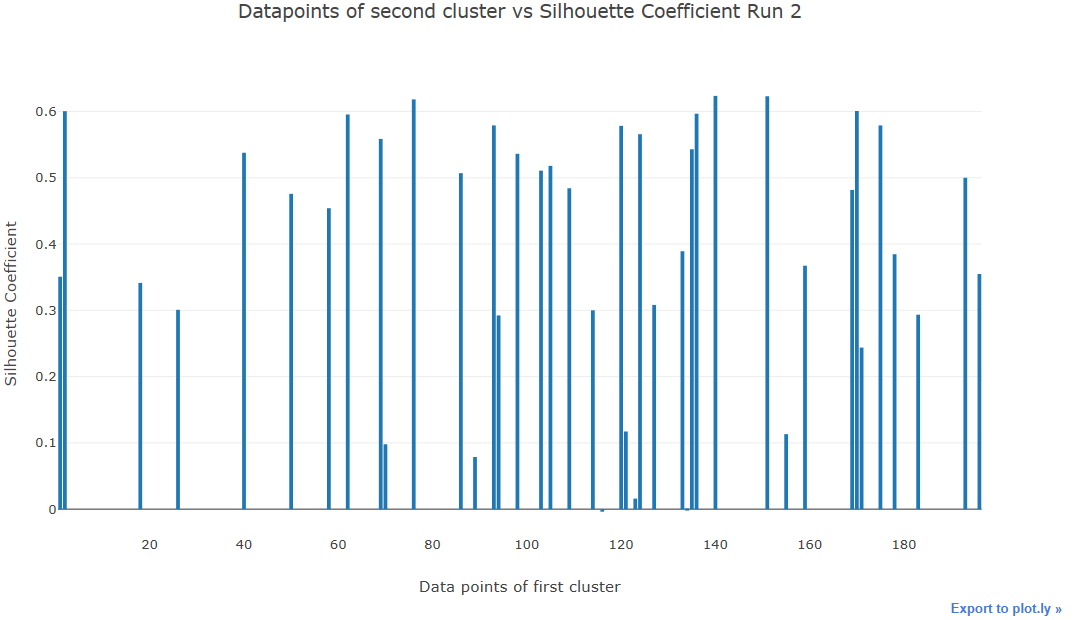


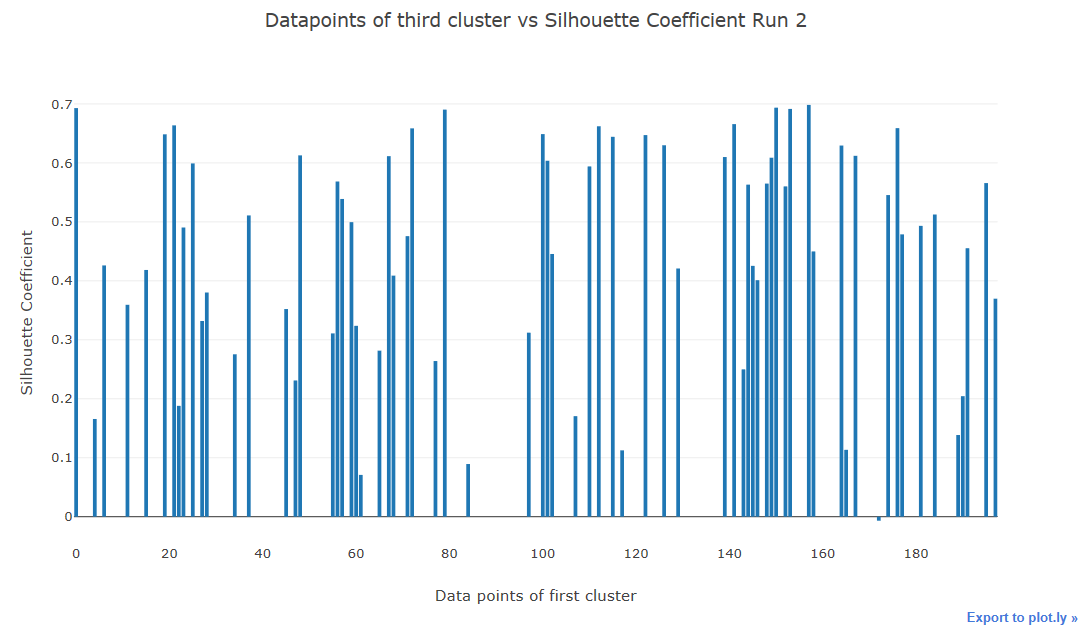
**Silhouette coefficient vs Data points**

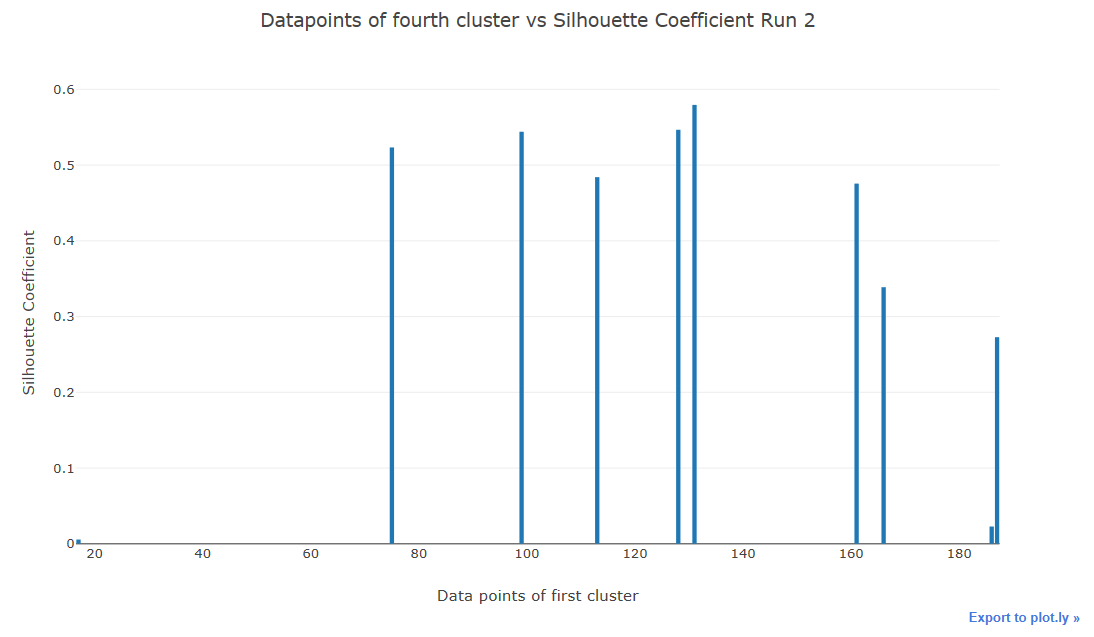




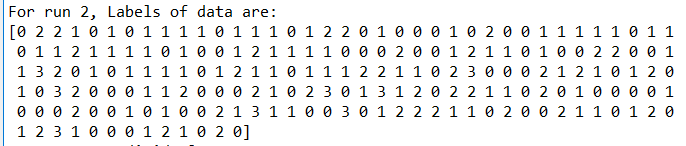


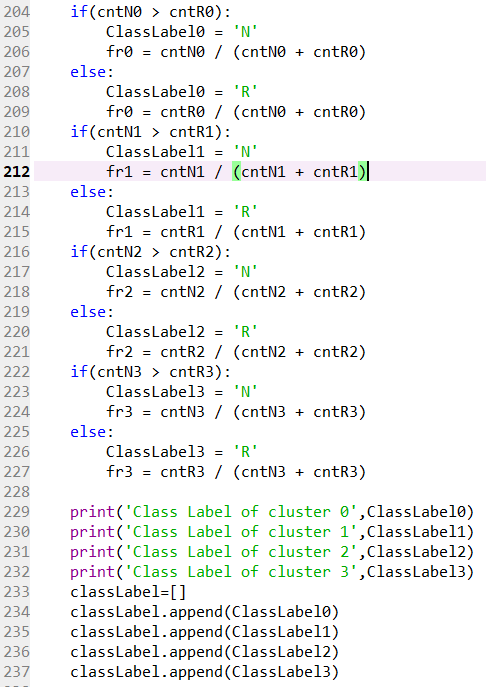




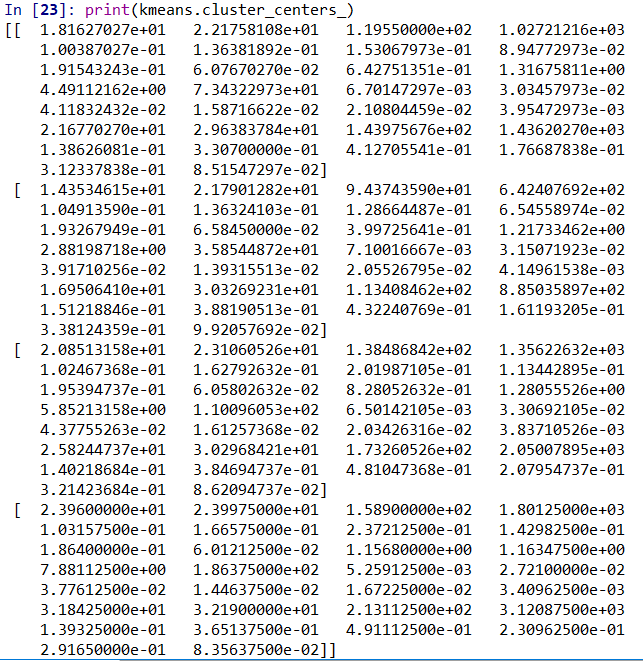


**Now considering the class label for each data point in each cluster (“R” or “N”). To each cluster assigning the label that belongs to most of the data points in that cluster. Reporting the cluster center, its SSE, and its class label, and the fraction of points that have the class label.**





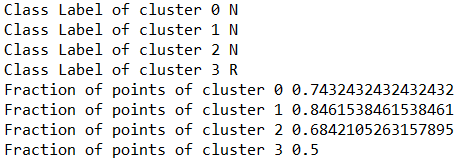
**Cluster center:**



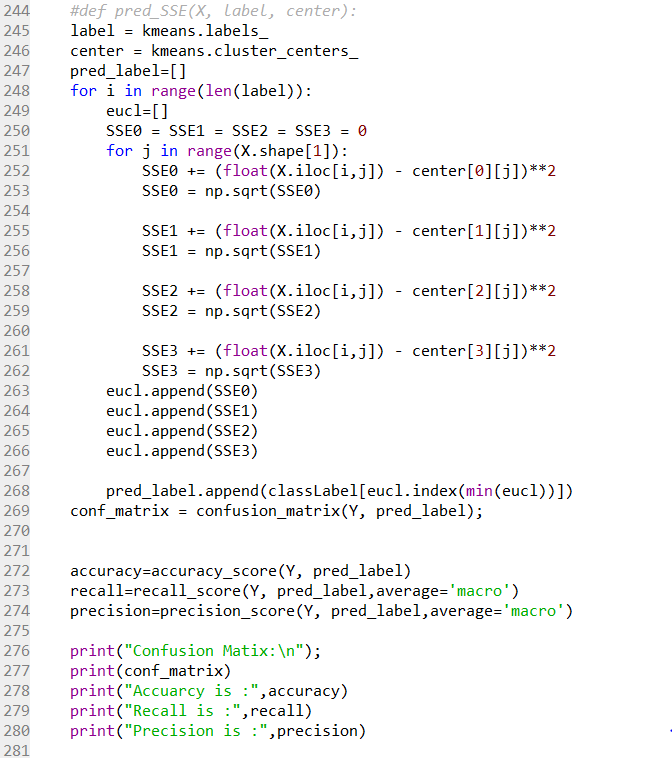
**SSE:**

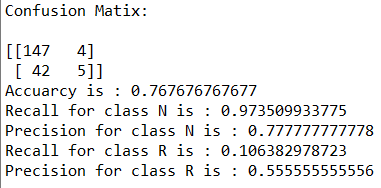


**Class labels and Fraction of points of cluster (run 2):**

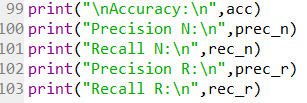


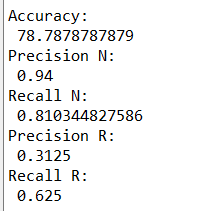
**Now, using the cluster centers and the class labels as a new classifier, considering each data point again as belonging to your test set. For each data point, predicting its class label to be the one that belongs to the cluster center that is closest to the data point. Building the confusion matrix for this new classifier and computing its accuracy, precision and recall values.**





**The precision and recall values of each of the two classes obtained using Decision Tree model.**





**Comparison based on performance measures between Decision tree result and Clustering:**

1. According to accuracy, Decision tree is slightly more accurate than the clustering. Decision tree is 78% accurate while the clustering is 76%.
2. The precision and recall values for Decision tree are better compared to the clustering.

**Reason for difference:**

**Clustering** techniques can group attributes into a few similar segments where data within each group is similar to each other and distinctive across groups. It is an unsupervised learning process finding logical relationships and patterns from the structure of the data.

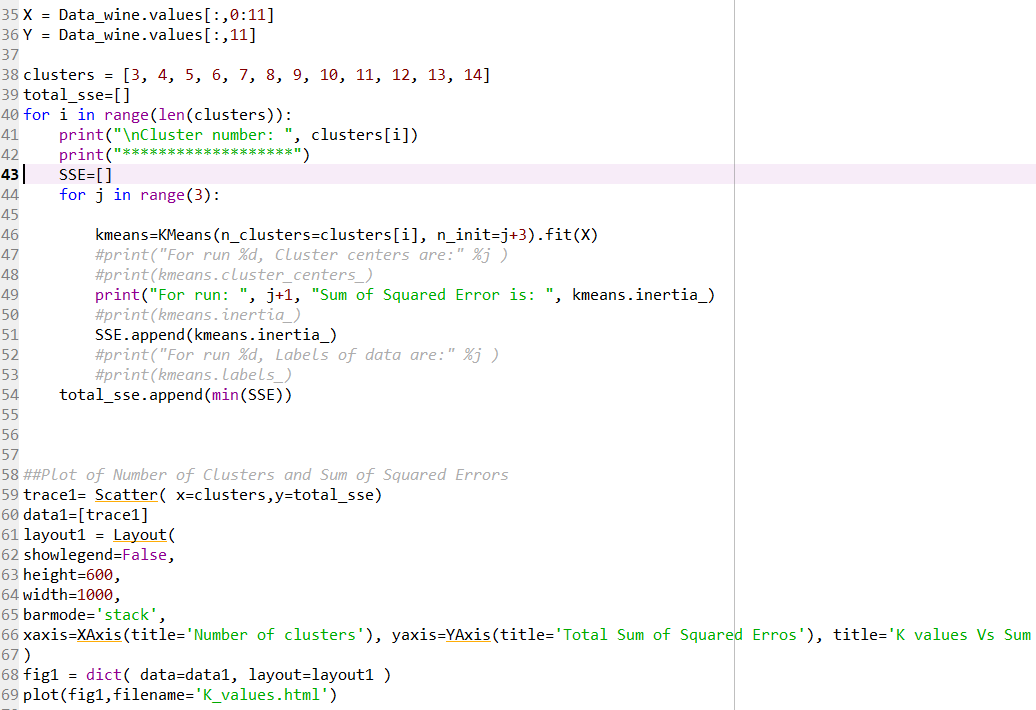
Discovers the underlying rules that collectively define a cluster.

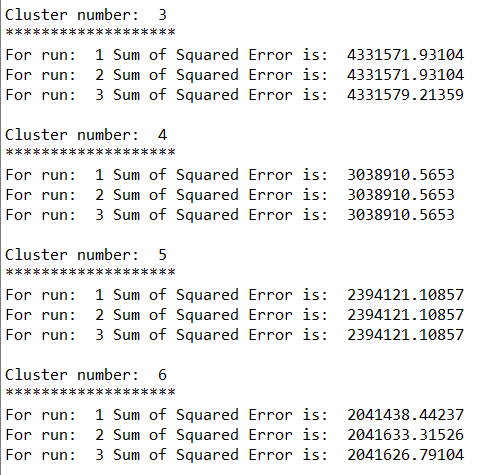
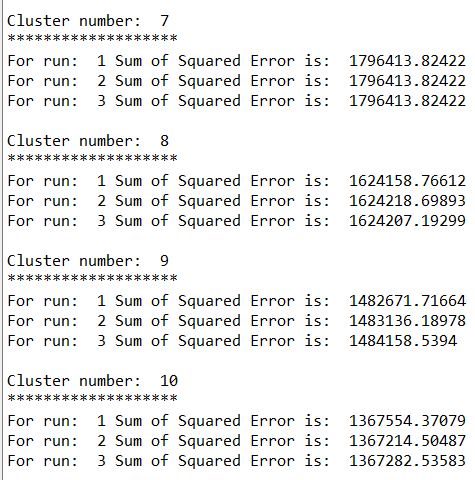
**Decision** **trees** are supervised learning techniques. They are arranged in a hierarchical tree-like structure and are simple to understand and interpret. Decision trees can be well-suited for cases in which we need the ability to explain the reason for a particular decision.

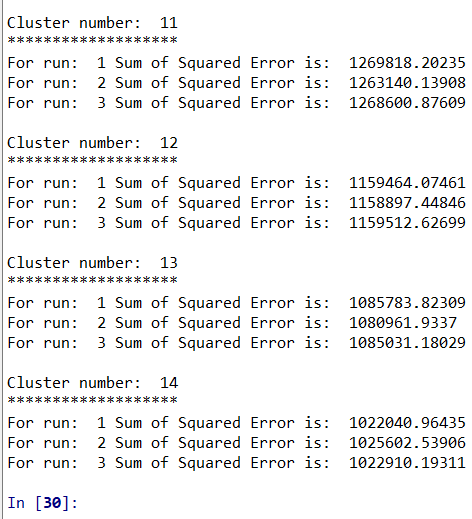
Classifies instances into known groups.

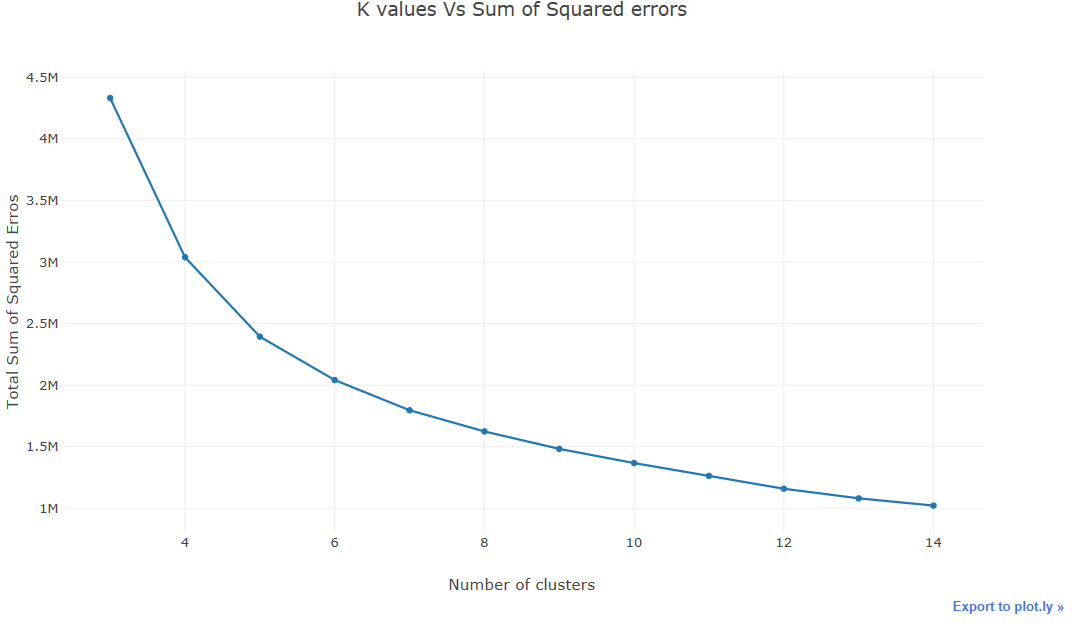
Having said that, the nature of data here is categorical and fits the requirements of the decision tree classifier more aptly than the clustering algorithm. Since, **decision tree can handle outliers better** than the clustering as clustering depends on the data itself, Decision tree performance is slightly better.

**Mixing the datasets for the red and white wines in one dataset. Performing k-means clustering on this large dataset for the values of k to be: 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, and 14. For each value of k reporting the lowest total SSE value after selecting the best of the 3-runs for each value of k. Plotting the SSE value vs. the value of k.**







From the above plot, it can be inferred that the sum of squared error decreases with increase in cluster number. This is because, as the number of clusters increases, the proportion of region covered by each cluster significantly increases there by minimizing the misclassification of the instances, Thus, the probability of the instance falling into the right cluster increases.

This can be better understood by exaggerating it further to the extent that as the number of clusters increases so much so that each cluster holds exactly single instance that is labeled. Thus, the error is significantly reduced.