Implementation of k-means clustering algorithm

Abhinav

Introduction

The objective is to implement a k-means clustering algorithm and test its performance on the given three datasets, namely two dimensional easy, two dimensional hard and wine quality. The performance of the algorithm is compared against the performance of off-the-shelf clustering algorithms. The report aims to cover the following phases of the project :

1. Model description and design choices
2. Performance evaluation
3. Comparison with off-the-shelf implementation

Model description and design choices

Data processing

The first step was understanding the data sets and choosing the appropriate techniques to preprocess the data before implementing the clustering algorithm. All three data sets in consideration consisted primarily of numerical features, with the exception of feature “id” in all data sets and the feature “class” in the wine quality dataset. The feature id was converted to an integer and the feature “class” was enumerated with values 1 for “low” and 2 for “high”. This enabled input data to be converted into a numerical matrix which increased the ease of performing calculations and manipulations. Both features in the two dimensional easy and hard data sets were of similar range (~ -0.5 to 1.5), while the features in the wine quality set had huge variations. Even though normalization was not necessary for the first two datasets, min-max normalization was performed on all datasets to bring all

features to the same class and thereby provide equal weightage to all features. The class features and quality (in wine set) were not considered for the clustering process.

K-means implementation

The next part was the implementation of the k-means clustering algorithm. To start off the process, k random points from the input dataset were chosen as k cluster centroids and then each point in the dataset was assigned a cluster number using the euclidean distance measure. Then the centroids for each cluster is updated to the mean of all the data points that were assigned to that particular cluster. During the cluster assignment process, it is possible that a cluster might not be assigned to any points which leads to a problem in updation of that cluster's centroid. The occurrence of empty clusters were handled by assigning the data point with the highest cluster sse as the new centroid of the cluster, which provides a better starting point for finding the actual cluster.

Performance Evaluation

The performance of the clustering algorithm was assessed using multiple measures such as the cluster sum of squared errors (SSE), between-cluster sum of squared errors (SSB) and average silhouette widths. Further observations about the performance of the algorithm were noted using scatter plots and confusion matrices. The evaluation report is divided into three parts, each describing the performance of one particular dataset, specifically the analysis of i) two dimensional easy dataset, ii) two dimensional hard dataset, and iii) wine quality dataset.

K-means clustering on Two Dimensional Easy data

From the images shown below (*fig 2.1*) it can be seen that the estimated squared error values for the Easy dataset with k = 2 is slightly better than the true values of the dataset. The overall estimated cluster SSE is slightly lesser indicative of better intra cluster cohesion and the total SSB is slightly greater indicative of greater cluster separation .

```
True SSE for TwoDimEasy
Total Cluster SSE 7.6582681305
Cluster 0 SSE 0.934313563554
Cluster 1 SSE 6.72395456695
Total SSE 38.1118255121
SSB 30.4535573816

Estimated SSE for TwoDimEasy k = 2
Total Cluster SSE 7.55135705534
Cluster 0 SSE 1.10134235227
Cluster 1 SSE 6.45001470307
Total SSE 38.1118255121
SSB 30.5604684568

Estimated SSE for TwoDimEasy k = 3
Total Cluster SSE 5.38810998624
Cluster 0 SSE 2.38092081765
Cluster 1 SSE 1.01454243551
Cluster 2 SSE 1.99264673307
Total SSE 38.1118255121
SSB 32.7237155259

Silhouette width for TwoDimEasy k = 2
Silhoutte Coefficient 0 0.829341705865
Silhoutte Coefficient 1 0.595813598161
Total Silhoutte Coefficient 0.704793381756

Silhouette width for TwoDimEasy k = 3
Silhoutte Coefficient 0 0.226536161383
Silhoutte Coefficient 1 0.811485948099
Silhoutte Coefficient 2 0.369805420659
Total Silhoutte Coefficient 0.541021237875
```

Fig 2.1

It can also be seen from the silhouette coefficient for k=2 is quite high, which indicates good clustering. A silhouette coefficient value close to 1 means that the clustering has high cohesion within clusters and high measure of separation among the clusters.

K = 2 **K = 3**

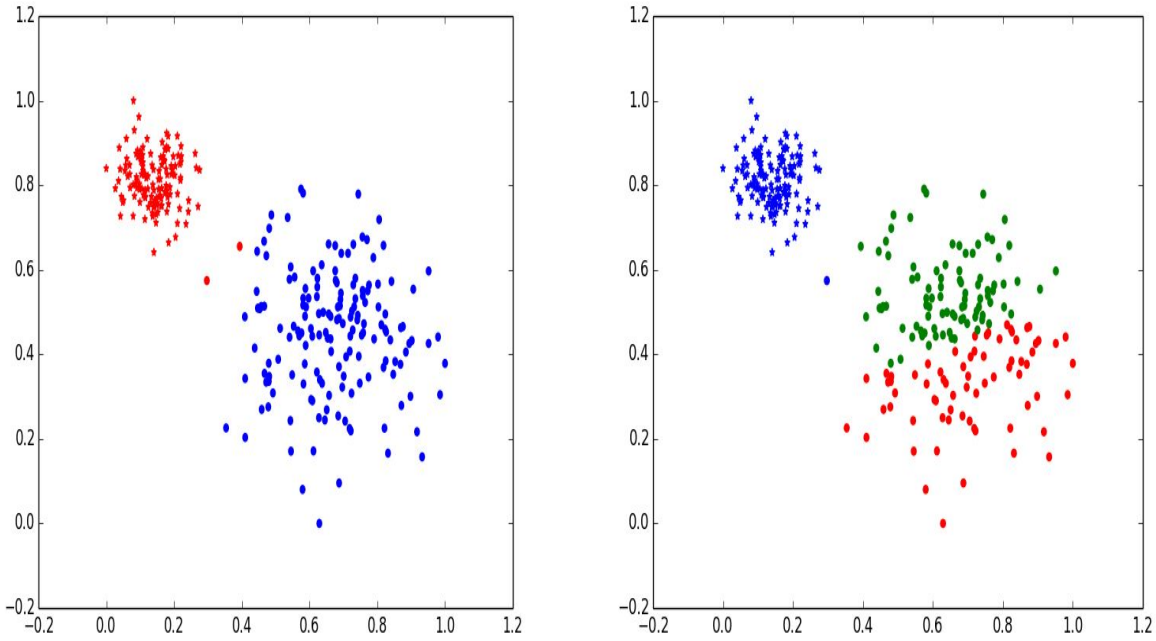


Fig 2.2

The scatter plot displayed above in fig 2.2 show the high degree of overlap between the original cluster classes (stars and circles) and the estimated clusters (red and blue). The extremely low level of misclassification can be observed in the plot and the confusion matrix (*fig 2.3*) shown below.

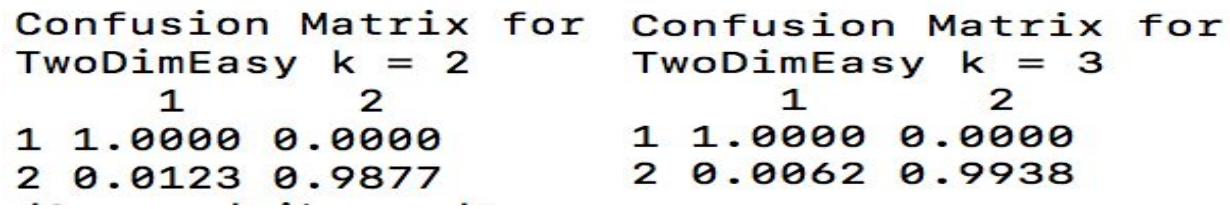


Fig 2.3

It was further observed that for the value of k=2, the metrics for two dimensional easy data remained constant irrespective of the value of the initial centroids. The SSE values for three runs with random initial values can be seen below in fig 2.4. This denotes that the final clusters and their centroids are the same for this particular dataset and value of k irrespective of the starting centroids.

Run 1

```
Estimated SSE for TwoDimEasy k = 2
Total Cluster SSE 7.55135705534
Cluster 0 SSE 1.10134235227
Cluster 1 SSE 6.45001470307
Total SSE 38.1118255121
SSB 30.5604684568
```

Run 2

```
Estimated SSE for TwoDimEasy k = 2
Total Cluster SSE 7.55135705534
Cluster 0 SSE 1.10134235227
Cluster 1 SSE 6.45001470307
Total SSE 38.1118255121
SSB 30.5604684568
```

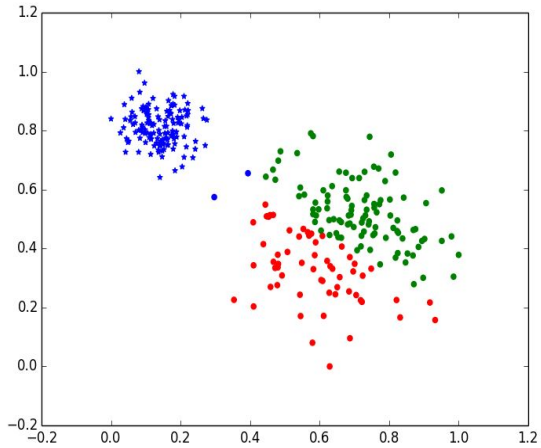
Run 3

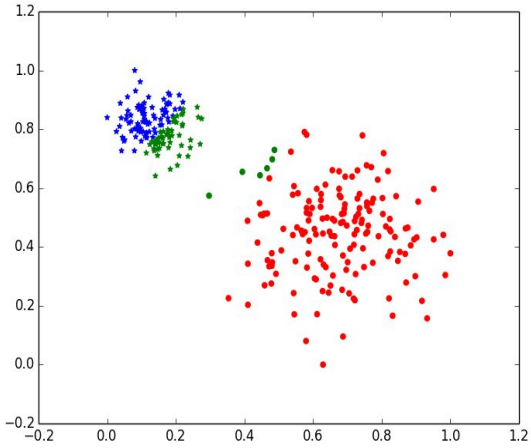
```
Estimated SSE for TwoDimEasy k = 2
Total Cluster SSE 7.55135705534
Cluster 0 SSE 1.10134235227
Cluster 1 SSE 6.45001470307
Total SSE 38.1118255121
SSB 30.5604684568
```

Fig 2.4

For the value of k=3, it can be seen (*fig 2.1*) that the total cluster SSE decreases when compared to k=2 conveying better intra cluster cohesion but it can also be seen that the silhouette coefficient actually decreases which means that the clustering is actually worse than k=2. Also, although the scatter plot (*fig 2.2*) and the confusion matrix (*fig 2.3*) seem to indicate better classification, it was observed that multiple runs of the algorithm gave classification results both better and worse than k=2. The scatter plot and confusion matrices can be seen below (*fig 2.5*).

Run 1 **Run 2**





```
Confusion Matrix for
TwoDimEasy k = 3
      1       2
1 1.0000 0.0000
2 0.0123 0.9877
```

```
Confusion Matrix for
TwoDimEasy k = 3
      1       2
1 1.0000 0.0000
2 0.0370 0.9630
```

Fig 2.5

K-means clustering on Two Dimensional Hard data

For multiple runs of k as 4, we can observe that the total estimated cluster SSE was considerably lower, 6.78 as opposed to 7.69 for the true clustering, which suggests that the clustering produced by the k-means algorithm was better than the true clustering.

Run 1 **Run 2** **Run 3**

```
True SSE for TwoDimHard
Total Cluster SSE 7.69628580125
Cluster 0 SSE 0.435498632431
Cluster 1 SSE 1.25877334472
Cluster 2 SSE 3.35053610917
Cluster 3 SSE 2.65147771493
Total SSE 40.6680268451
SSB 32.9717410439
```

```
Estimated SSE for TwoDimHard k = 4
Total Cluster SSE 6.78038761885
Cluster 0 SSE 2.50346668965
Cluster 1 SSE 1.49663929573
Cluster 2 SSE 2.08853926057
Cluster 3 SSE 0.69174237291
Total SSE 40.6680268451
SSB 33.8876392263
```

```
Estimated SSE for TwoDimHard k = 4
Total Cluster SSE 6.77944484822
Cluster 0 SSE 2.04197718063
Cluster 1 SSE 1.49663929573
Cluster 2 SSE 2.54908599895
Cluster 3 SSE 0.69174237291
Total SSE 40.6680268451
SSB 33.8885819969
```

```
Estimated SSE for TwoDimHard k = 4
Total Cluster SSE 6.78038761885
Cluster 0 SSE 0.69174237291
Cluster 1 SSE 1.49663929573
Cluster 2 SSE 2.08853926057
Cluster 3 SSE 2.50346668965
Total SSE 40.6680268451
SSB 33.8876392263
```

Fig 2.6

Run 1

```
Silhouette width for TwoDimHard k = 4
Silhoutte Coefficient 0 0.450216963135
Silhoutte Coefficient 1 0.51427413768
Silhoutte Coefficient 2 0.516539837394
Silhoutte Coefficient 3 0.695972133501
Total Silhoutte Coefficient 0.543786429274
```

Run 2

```
Silhouette width for TwoDimHard k = 4
Silhoutte Coefficient 0 0.521091251323
Silhoutte Coefficient 1 0.514000884386
Silhoutte Coefficient 2 0.445582714819
Silhoutte Coefficient 3 0.696054852425
Total Silhoutte Coefficient 0.543741286798
```

Run 3

```
Silhouette width for TwoDimHard k = 4
Silhoutte Coefficient 0 0.695972133501
Silhoutte Coefficient 1 0.51427413768
Silhoutte Coefficient 2 0.516539837394
Silhoutte Coefficient 3 0.450216963135
Total Silhoutte Coefficient 0.543786429274
```

Fig 2.7

The silhouette coefficient for two dimensional hard dataset was observed to be approximately 0.5437 across multiple runs, which shows that the clustering was fairly decent in terms of intracluster cohesion and intercluster separation. The scatter plot for the first run (*fig 2.8*) and the confusion matrices for all the runs (*fig 2.9*) show that the estimated clustering was fairly representative of the original clustering across runs with occasional misclassification errors at junctions of two cluster boundaries. It can also be observed from the plot and matrix that the cluster at the top left is a tightly bound cluster with a true positive of 1 in all the runs.

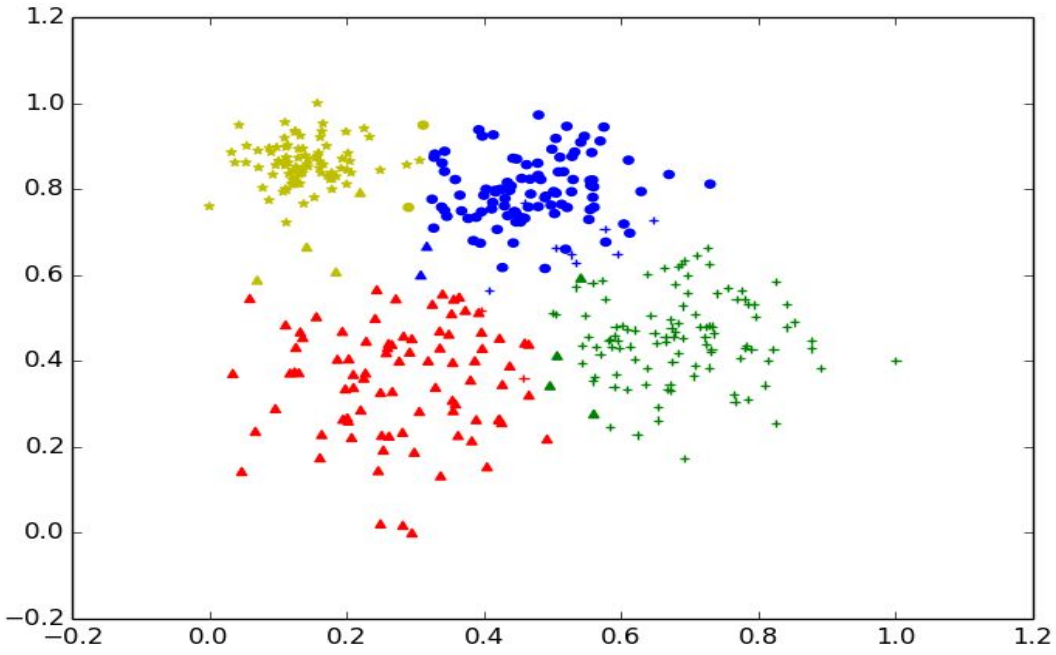


Fig 2.8

Run 1 **Run 2** **Run 3**

Confusion Matrix for TwoDimHard k = 4

	1	2	3	4
1	1.0000	0.0000	0.0000	0.0000
2	0.0200	0.9800	0.0000	0.0000
3	0.0412	0.0206	0.8969	0.0412
4	0.0000	0.0702	0.0175	0.9123

Confusion Matrix for TwoDimHard k = 4

	1	2	3	4
1	1.0000	0.0000	0.0000	0.0000
2	0.0200	0.9800	0.0000	0.0000
3	0.0412	0.0206	0.9072	0.0309
4	0.0000	0.0702	0.0175	0.9123

Confusion Matrix for TwoDimHard k = 4

	1	2	3	4
1	1.0000	0.0000	0.0000	0.0000
2	0.0200	0.9800	0.0000	0.0000
3	0.0412	0.0206	0.8969	0.0412
4	0.0000	0.0702	0.0175	0.9123

Fig 2.9

From the figures shown above (*fig 2.6, 2.7, 2.8, and 2.9*) it can be seen that although multiple runs of k-means for k=4 for two dimensional hard dataset result in different clusterings, the difference is negligible and the various metrics seem to remain consistent across runs. When the value of k is changed to 3, the performance of the clustering algorithm is noted to be worse (*fig 2.10*). The total cluster SSE doubled, the silhouette coefficient reduced pointing to a poorer performance. The high misclassification error rate observed is unsurprising given that it is fitting 4 original clusters into 3 new ones.

Estimated SSE for TwoDimHard k = 3
Total Cluster SSE 14.6693786007
Cluster 0 SSE 2.14996702379
Cluster 1 SSE 10.8986024604
Cluster 2 SSE 1.62080911645
Total SSE 40.6680268451
SSB 25.9986482445

Silhouette width for TwoDimHard k = 3
Silhoutte Coefficient 0 0.49172808298
Silhoutte Coefficient 1 0.289922039627
Silhoutte Coefficient 2 0.629600761956
Total Silhoutte Coefficient 0.43607289661

Confusion Matrix for TwoDimHard k = 3

	1	2	3	4
1	1.0000	0.0000	0.0000	0.0000
2	0.0200	0.9800	0.0000	0.0000
3	0.1134	0.0309	0.0000	0.8557
4	0.0000	0.1491	0.0000	0.8509

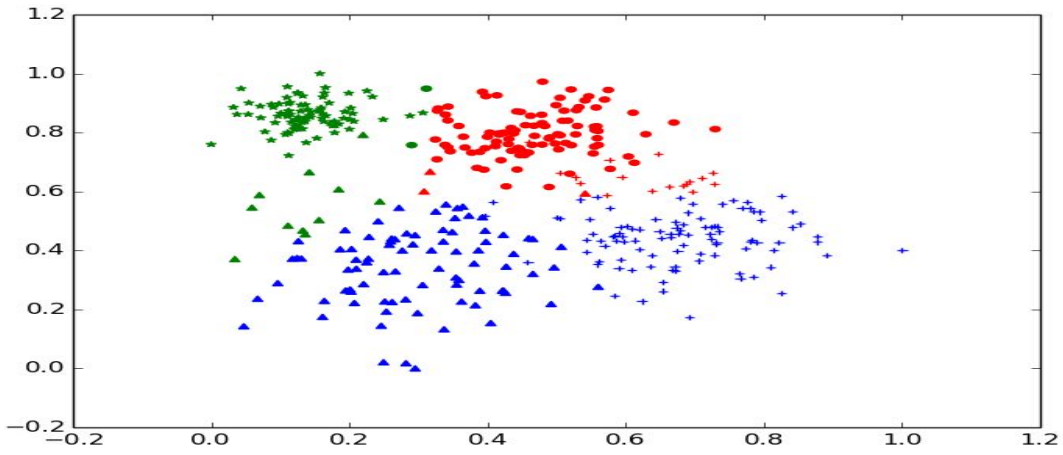


Fig 2.10

K-means clustering on Wine Quality data

```
Estimated SSE for wine_quality-red k = 2
Total Cluster SSE 239.183110299
Cluster 0 SSE 121.908843279
Cluster 1 SSE 117.274267019
Total SSE 317.373388474
SSB 78.1902781758

Estimated SSE for wine_quality-red k = 3
Total Cluster SSE 209.336489845
Cluster 0 SSE 77.6210424884
Cluster 1 SSE 85.0079799559
Cluster 2 SSE 46.707467401
Total SSE 317.373388474
SSB 108.036898629

Estimated SSE for wine_quality-red k = 4
Total Cluster SSE 183.868433846
Cluster 0 SSE 52.6538997045
Cluster 1 SSE 49.1122289742
Cluster 2 SSE 37.835162532
Cluster 3 SSE 44.2671426356
Total SSE 317.373388474
SSB 133.504954628
```

```
Silhouette width for wine_quality-red k = 2
Silhoutte Coefficient 0 0.296141171007
Silhoutte Coefficient 1 0.171040158743
Total Silhoutte Coefficient 0.244504730673

Silhouette width for wine_quality-red k = 3
Silhoutte Coefficient 0 0.17968226578
Silhoutte Coefficient 1 0.264452984098
Silhoutte Coefficient 2 0.14932821259
Total Silhoutte Coefficient 0.215973689573
```

Fig 2.11

For the performance analysis of the wine quality dataset, the SSE, silhouette coefficient, and the confusion matrix were computed for various values of k. As a measure of checking the impurity of the clusters, the Gini index, mean quality and standard deviation of the quality of each cluster were computed using the quality attribute provided in the dataset. The quality features contained ordinal values 3 - 8 with a majority of the values (80%) having 5 or 6. The data were categorized into two classes, high if quality is 6 or more and low otherwise. The output of the analysis for k values of 2, 3, and 4 can be seen in the images shown above and below (*fig 2.11 and 2.12*).

K = 2

```
Cluster Numbers          0          1
Cluster Gini Index [ 0.49444212  0.44938017]
Cluster Qual Mean  [ 5.45899894  5.88787879]
Cluster Qual StDev [ 0.73347718  0.84000634]
Total Gini 0.475842438556
```

```
Confusion Matrix for
wine_quality-red k = 2
      1      2
1 0.6976 0.3024
2 0.4912 0.5088
```

K = 3

```
Cluster Numbers          0          1          2
Cluster Gini Index [ 0.45355278  0.27107462  0.46364739]
Cluster Qual Mean  [ 5.31108312  6.14371257  5.82377919]
Cluster Qual StDev [ 0.64873687  0.76819603  0.82987436]
Total Gini 0.418410098985
```

```
Confusion Matrix for
wine_quality-red k = 3
      1      2
1 0.6935 0.3065
2 0.3345 0.6655
```

```
Confusion Matrix for
wine_quality-red k = 4
      1      2
1 0.7782 0.2218
2 0.3942 0.6058
```

Fig 2.12

It can observed from the plots shown below that as the value of k increases the value of SSE, Silhouette coefficient and the Gini index decreases. The decrease in SSE can be explained by the fact that increase in number of clusters would lead to smaller and closely packed clusterings. This could be a good result but it must also be noted that the Silhouette coefficient also reduces leading to the conclusion that the cluster separation is not very good and the clustering structure becomes poorer as k increases. It can also be observed that the value of Gini index reduces as the value of k increases which indicates a decrease in classification error reaching a value of 0.38 which is still considered to be high.

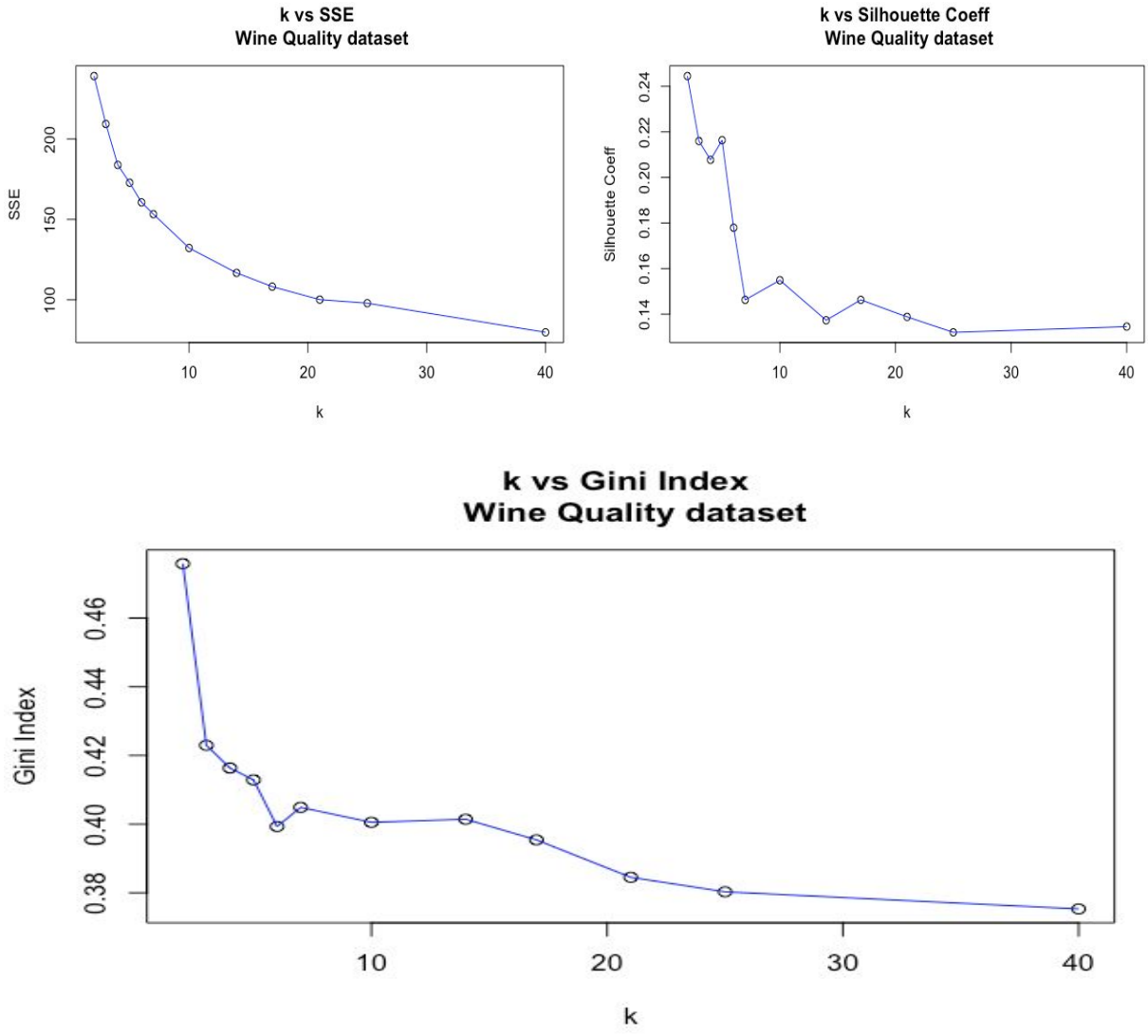


Fig 2.13

The value of the average quality of the cluster ranged between 5 to 6.5 for all values of k owing to most points have a quality of 5 or 6, and the value of standard deviation reduced slightly with increase in k. To choose the best value of k, the metrics silhouette coefficient and gini index were considered since silhouette coefficient provides a good measure of the cluster structure and its cohesion and gini index provides insights on the cluster impurity value (classification uncertainty and error).

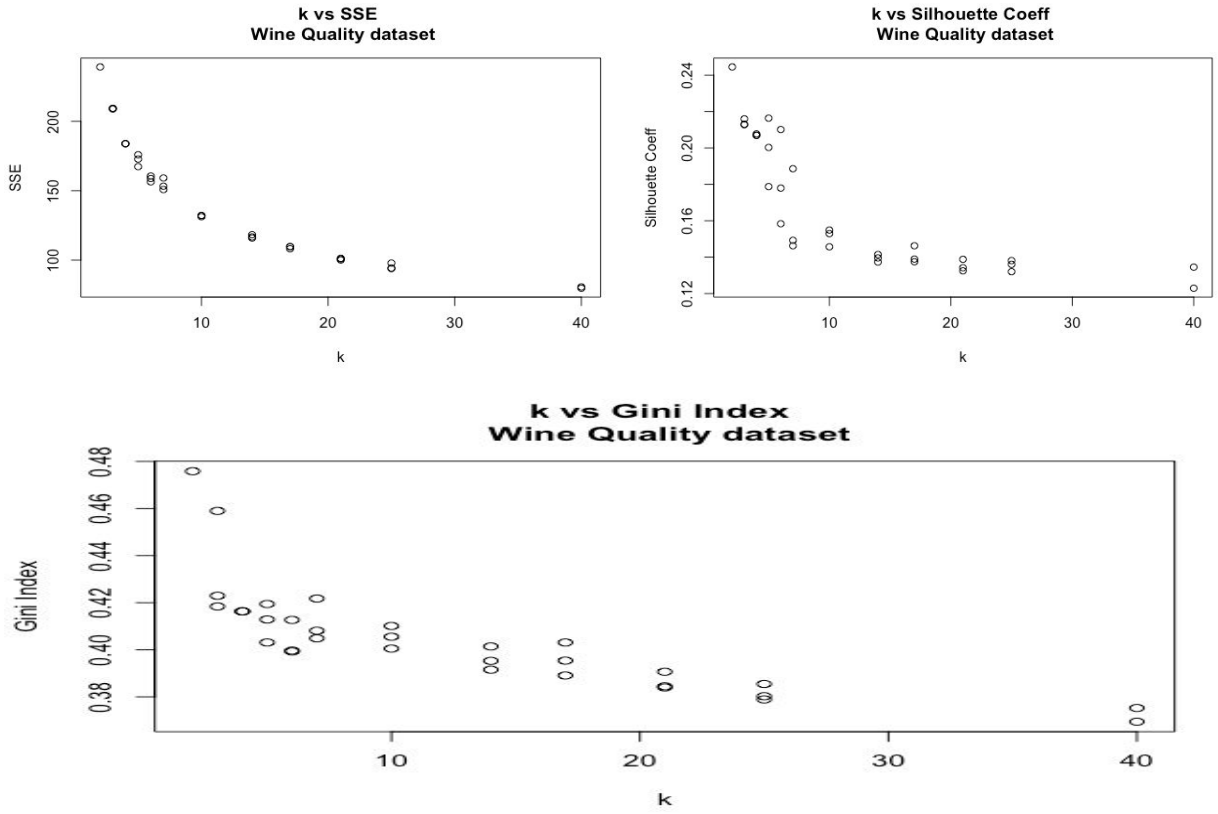


Fig 2.14

The datasets were run multiple times for each value of k to ensure that the actual performance at that k is captured. It can be seen from fig 2.14 that although the pattern follows the same trends that were observed in fig 2.13, there is quite some variance in the performance of the smaller values of k. To find the best value of k there has to be a trade off between cohesion and impurity. It can be seen that the value of gini index begins to level off after the value of k as 10. The best value of k for this cluster would be 5 or 6 where the gini index (0.4) is comparable to the saturation (0.38) and the silhouette coefficient (0.2) is comparably high.

Comparison with off-the-shelf clustering methods

Off-the-shelf model description

The performance of the k-means clustering algorithm was compared against the off-the-shelf k-means algorithm provided by the scikit-learn package. The parameter n_init was set to default which returns the best output out of 10 runs. The initialization method was set to kmeans++ which provides initial points in such a way to speed up convergence. The k-means clustering algorithm was then used to fit the points in the three given dataset after processing the data using the same techniques.

Performance Comparison

Two dimensional easy dataset

It can be observed from the results shown below (*fig 3.1*) that the performance of the off-the-shelf clustering algorithm for the two dimensional dataset is identical to that of the custom built one for the value of k =2.

```
Estimated SSE for TwoDimEasy k = 2
Total Cluster SSE 7.55135705534
Cluster 0 SSE 6.45001470307
Cluster 1 SSE 1.10134235227
Total SSE 38.1118255121
SSB 30.5604684568
```

```
Silhouette width for TwoDimEasy k = 2
Silhoutte Coefficient 0 0.595813598161
Silhoutte Coefficient 1 0.829341705865
Total Silhoutte Coefficient 0.704793381756
```

```
Confusion Matrix for
TwoDimEasy k = 2
      1      2
1 1.0000 0.0000
2 0.0123 0.9877
```

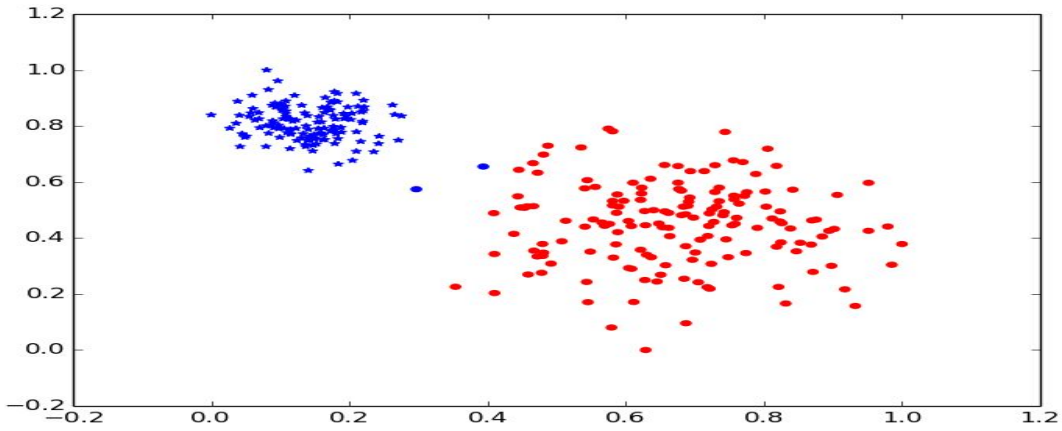


Fig 3.1

Two dimensional hard dataset

Though the performance of the off-the-shelf algorithm in the two dimensional hard dataset is not exactly the same as the custom algorithm, they are almost similar with minute differences. For the value of k as 4, the total Cluster SSE and the silhouette width can be seen to approximately the same with a difference of 0.001 in their values. The confusion matrix displays better results in the off-the-shelf algorithm with class 3 having a better true positive and the remaining classes maintaining the same values. Therefore it can be concluded that the performance of the off-the-shelf clustering algorithm is better by a very small margin that it could be considered the same.

Confusion Matrix for TwoDimHard k = 4

	1	2	3	4
1	1.0000	0.0000	0.0000	0.0000
2	0.0200	0.9800	0.0000	0.0000
3	0.0412	0.0103	0.9175	0.0309
4	0.0000	0.0702	0.0175	0.9123

Silhouette width for TwoDimHard k = 4
Silhoutte Coefficient 0 0.52146208213
Silhoutte Coefficient 1 0.520965547841
Silhoutte Coefficient 2 0.696179086101
Silhoutte Coefficient 3 0.439721375383
Total Silhoutte Coefficient 0.544228536866

Estimated SSE for TwoDimHard k = 4
Total Cluster SSE 6.77910059292
Cluster 0 SSE 1.43586772531
Cluster 1 SSE 2.04197718063
Cluster 2 SSE 0.69174237291
Cluster 3 SSE 2.60951331408
Total SSE 40.6680268451
SSB 33.8889262522

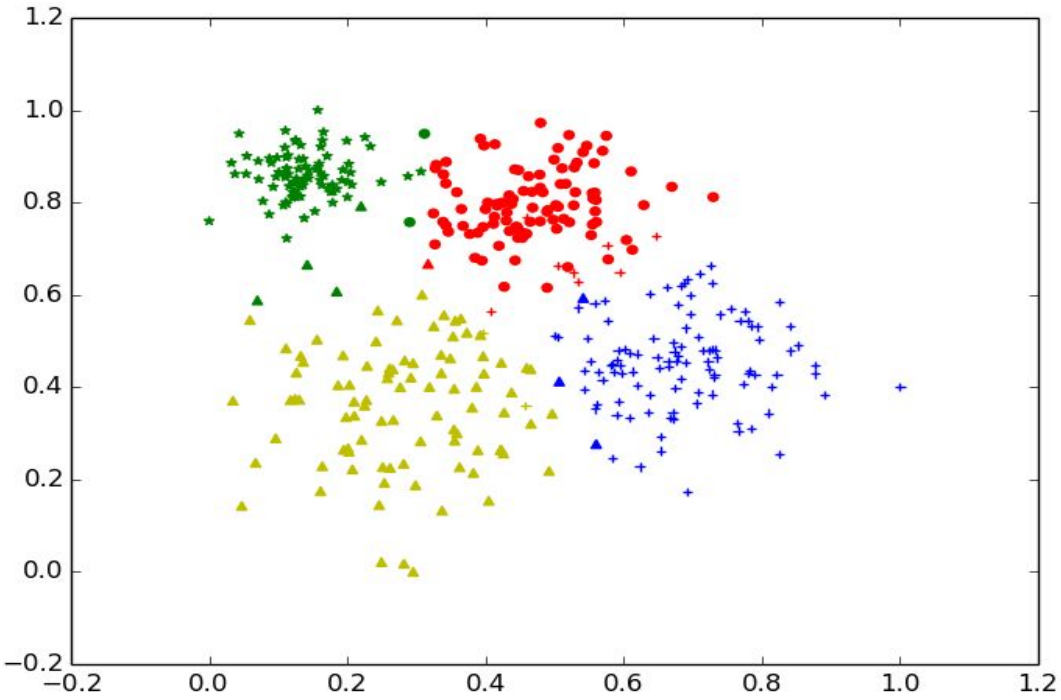


Fig 3.2

Wine Quality dataset

The plots of k versus the metrics SSE, Silhouette coefficient and Gini Index can be seen below in fig 3.3. The plots follow the same general trend that was noted during the runs of the custom built algorithm, although it can be observed that the curve of the plot is smoother for the off-the-shelf algorithm. It can also be seen that using the same logic as before, we can conclude from the plots below that k = 6 would be an optimal choice for this dataset, given the low values of SSE and Gini index and the relatively high value of Silhouette coefficient.

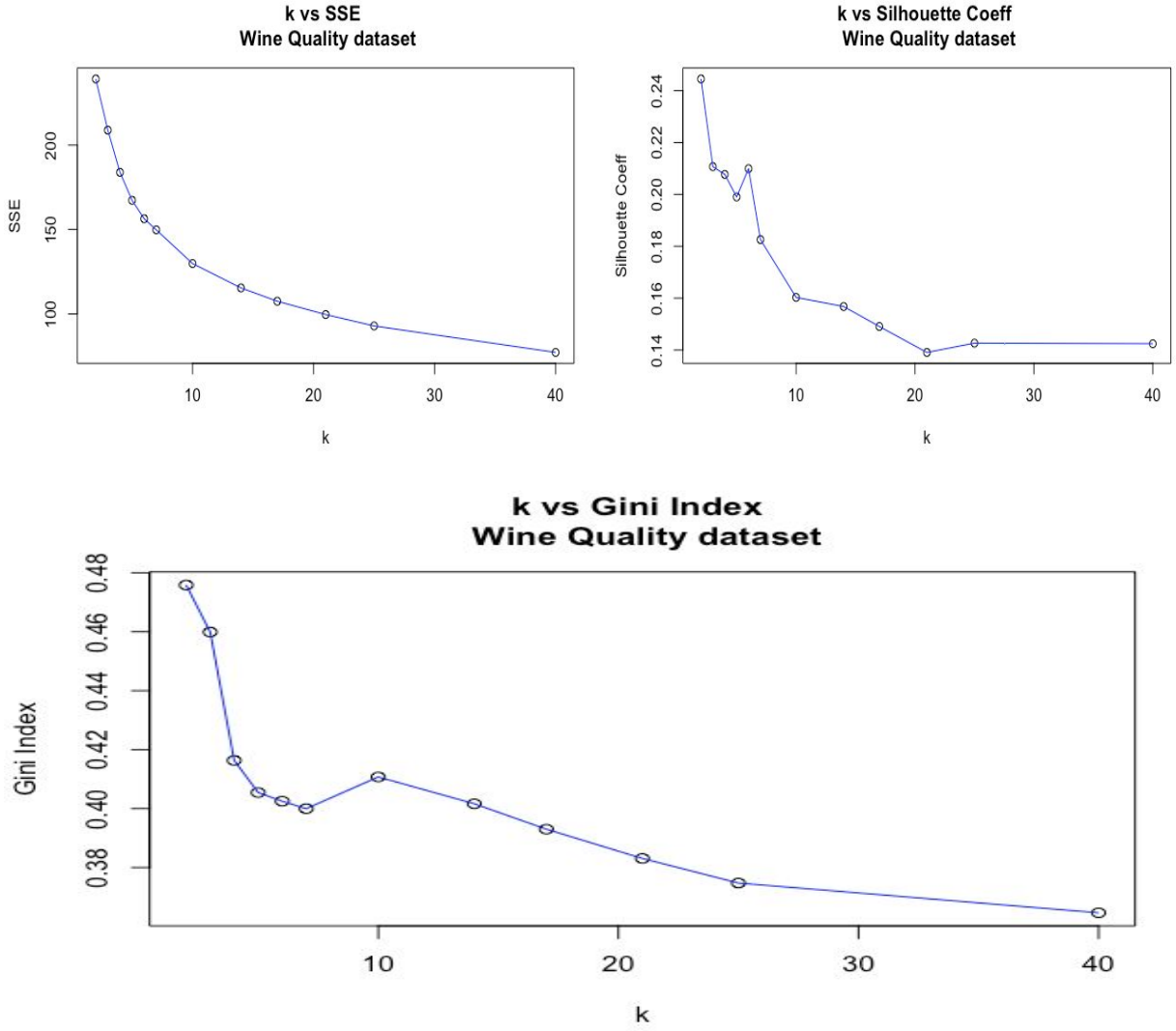


Fig 3.3