CSE 5243 - Data Mining Homework 4 Report

-Srividhya Chandrasekharan (chandrasekharan.12)

Part - 1

Section A

Introduction

The code *kmeans.py* is a Python program written to perform K-means clustering. The code takes in as input the dataset filename and the value of k(num. Of clusters), in that order.

Datasets used

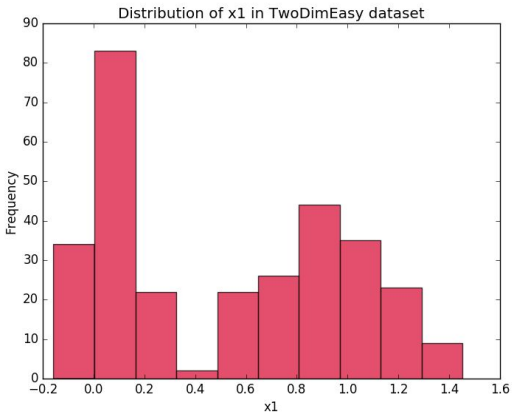
- TwoDimEasy.csv
- TwoDimHard.csv
- Wine_quality-red.csv

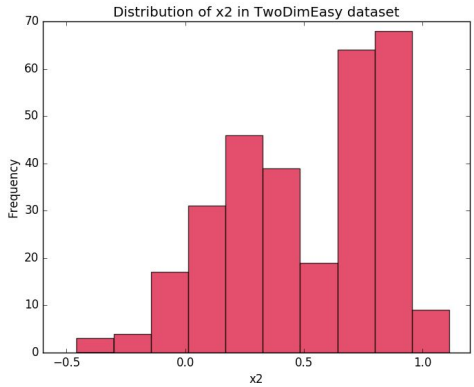
Data Preprocessing

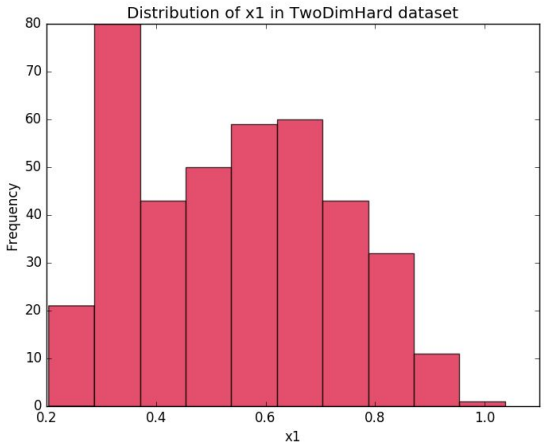
- Numeric attributes were normalized using min-max normalization so as to bring all the values to a common scale [0,1].

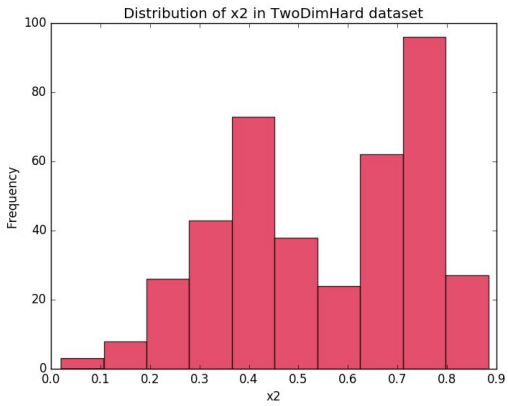
$$v' = \frac{v - min}{max - min}\ (max_{new} - min_{new}) + min_{new}$$

- By normalizing attributes, we make sure that different attributes contribute equally to the distance measures.
- No outliers were detected. Histogram was plotted to see the distribution of values and no peculiar trend was observed.









Algorithm used

- Choose k centroids
- Assign closest centroid to each data point
- Recompute centroids
- Reassign clusters. (repeat steps 2-4)

Objective function

Goal is to minimize the Sum of Squared Error(SSE)

Picking Initial Centroids

- Initial centroids are data points that are randomly picked from the collection of all data points. This is probably not the best way to pick the centroids, but this method works fine and the whole algorithm converges in small number of iterations.
- Even though the centroids are picked randomly, the final clusters' medians converge to the same set of centroids.

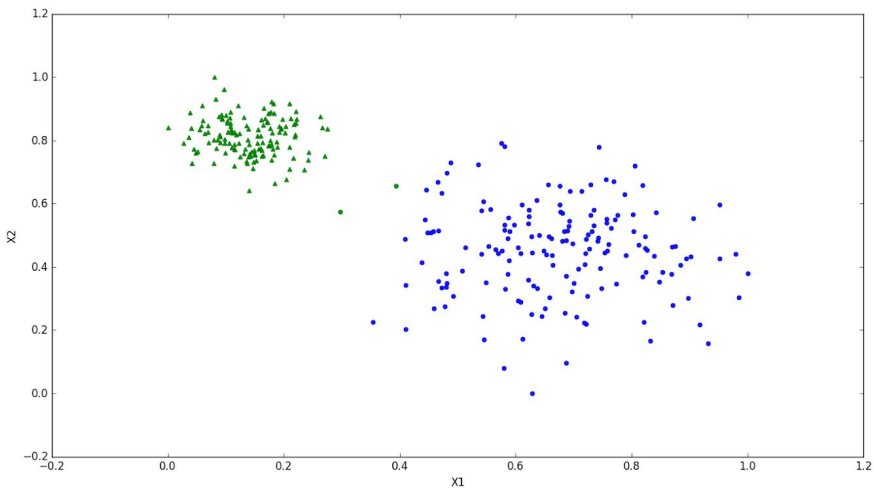
Clustering

- Euclidean distance between each datapoint and the k centroids is computed.
- Each centroid is considered as a cluster. So to start with, each cluster will have just one point i.e the centroid.
- Each data point is assigned to the cluster containing the closest centroid.
- After each pass of assigning all data points to a cluster, reestimation of cluster centroids happen. The new centroid has attribute values that are the mean of all the attribute values of the data points belonging to a particular cluster.
- This process is done iteratively until the centroid points don't change.

Handling empty clusters

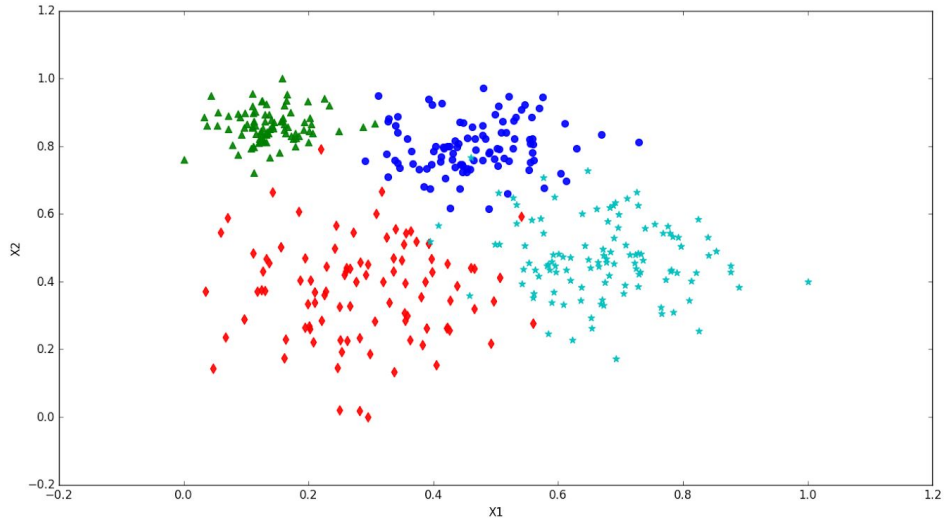
When no datapoint is assigned to a centroid, choose the point that's farthest from any current centroid and assign to the empty cluster.

Section B



The number of actual clusters in TwoDimEasy dataset is 2 and here's the visual representation of these clusters.

The number of actual clusters in TwoDimHard dataset is 4 and here's the visual representation of these clusters using a scatterplot against X1 and X2.



Formula used:-

- <u>True cluster sum of squared error(SSE)</u>: = $\sum_{i=1}^{k} \sum_{x \in Ci}$ distance^2(m_i,x)

  Where, C_i refers to the i^{th} cluster,
  m_i refers to the true centroid of the i^{th} cluster,
  x is a datapoint that belongs to the i^{th} cluster,
  k refers to the number of clusters,
  And, distance used is euclidean distance.

- <u>True overall Sum of squared error</u> = $\sum_{x \in C}$ distance^2(overall_m,x)

  Where: overall_m is the true overall centroid of all data points,
  x is each data point that's present in the dataset C,
  And, distance refers to the euclidean distance.

- True between-cluster sum of squares(SSB) = $\sum_{i=1}^{k} |C_i|$distance^2(m-mi)

  where, $|C_i|$ refers to the size of cluster C_i
  m is the true overall centroid of all data points,
  m_i refers to the centroid of the i^{th} cluster,
  and k refers to the number of clusters,

<u>INTERPRETING SCATTERPLOTS PRODUCED:</u>

- Each data point is plotted as a point using its X1 and X2 dimension values as x and y coordinates of the point, respectively.
- Each true cluster is assigned a shape. Markers used are triangle, circle, star, diamond.

- The cluster estimated for each point is represented using color. Colors used are red, green, blue and cyan.
- Misclassified points have a shape same as their true cluster and a color equal to the one assigned for their estimated cluster.

Analysis of TwoDimEasy.csv using True Clusters (4th attribute of data point)

- True centroid of cluster 1 : [1, 0.14117363738741998, 0.812058819633571]
- True centroid of cluster 2 : [2, 0.6689718538099726, 0.4513762488178289]

- Number of data points in cluster 1 is: 138
- Number of data points in cluster 2 is: 162

- SSE of cluster 1 = 0.9343135635537495
- SSE of cluster 2 = 7.658268130504978

- True overall SSE = 30.453557381589476

- Between-cluster sum of squares(SSB) = 0.11916609410187172

Analysis of TwoDimHard.csv using True Clusters (4th attribute of data point)

- True centroid of cluster 1 : [1, 0.13767349009778973, 0.8590885675868454]
- True centroid of cluster 2 : [2, 0.4649760301442808, 0.7981158762008937]
- True centroid of cluster 3 : [3, 0.28442072137442853, 0.3731321262822732]
- True centroid of cluster 4 : [4, 0.6713548962285426, 0.46913729315305996]

- Number of data points in cluster 1 is: 89
- Number of data points in cluster 2 is: 100
- Number of data points in cluster 3 is: 97
- Number of data points in cluster 4 is: 114

- SSE of cluster 1 = 0.43549863243055537
- SSE of cluster 2 = 1.6942719771487416
- SSE of cluster 3 = 5.044808086318167
- SSE of cluster 4 = 7.696285801246282

- True overall SSE = 32.971741043892074

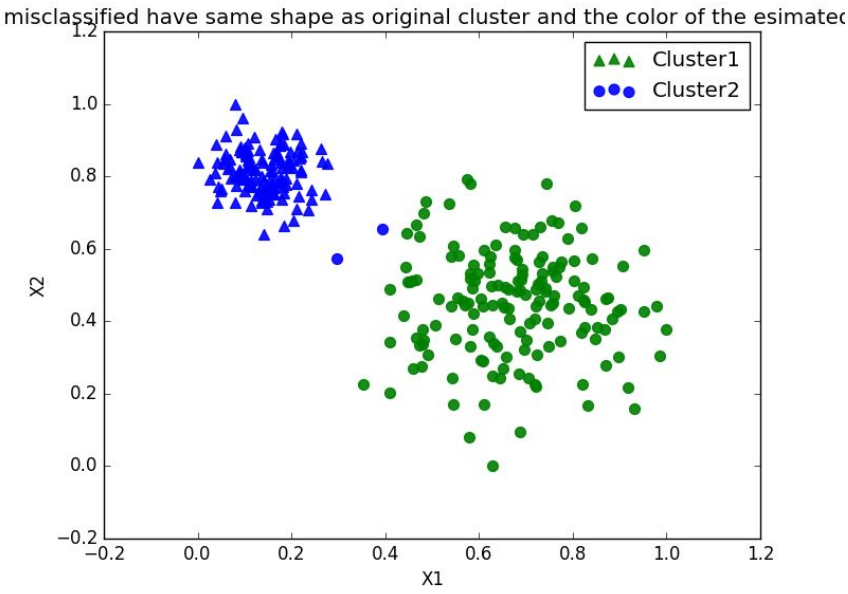
- Between-cluster sum of squares(SSB) = 0.13228142724133357

Section C

(C.1) On running the K means clustering algorithm for **3 times** on **TwoDimEasy** dataset for **k=2** and **different initial centroids**, the following results were obtained :-

(**Trial 1**)Initial Centroids :

	0	1	2	3
0	103	0.265815	0.840841	1
1	122	0.139236	0.748566	1



CONFUSION MATRIX

	1	2
1	138.0	0.0
2	2.0	160.0

Within cluster SSE of cluster 0 = 6.450014703071413
Within cluster SSE of cluster 1 = 1.10134235226528
Overall cluster SSE = 7.551357055336698
Between-cluster sum of squares(SSB) = 0.11642083221621999
Average Silhouette for cluster 0 = 0.256768428122
Average Silhouette for cluster 1 = 0.488881394088
Average Silhouette for entire dataset = 0.365087812239

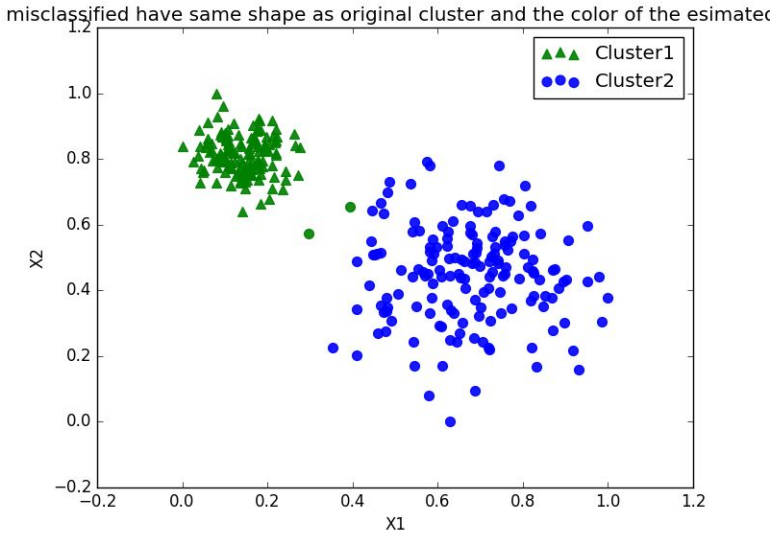
(Trial 2) Initial Centroids :

	0	1	2	3
0	66	0.110034	0.827333	1
1	99	0.156034	0.833357	1

CONFUSION MATRIX

	1	2
1	138.0	0.0
2	2.0	160.0

Within cluster SSE of cluster 0 = 1.10134235226528
Within cluster SSE of cluster 1 = 6.450014703071413
Overall cluster SSE = 7.551357055336692
Between-cluster sum of squares(SSB) = 0.11642083221621999
Average Silhouette for cluster 0 = 0.488881394088
Average Silhouette for cluster 1 = 0.256768428122
Average Silhouette for entire dataset = 0.365087812239



(Trial 3)

Initial Centroids :

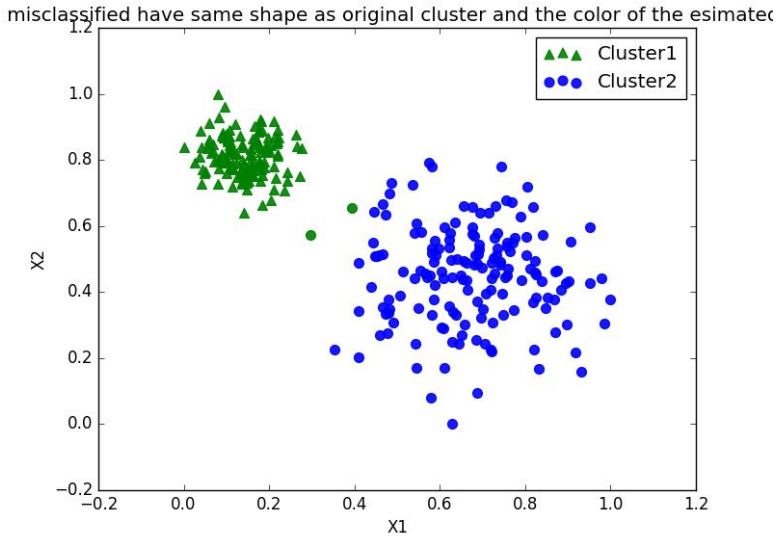
	0	1	2	3
0	11	0.000000	0.839712	1
1	85	0.176943	0.798755	1

CONFUSION MATRIX

	1	2
1	138.0	0.0
2	2.0	160.0

Within cluster SSE of cluster 0 = 1.10134235226528
Within cluster SSE of cluster 1 = 6.450014703071413
Overall cluster SSE = 7.551357055336692
Between-cluster sum of squares(SSB) = 0.11642083221621999
Average Silhouette for cluster 0 = 0.488881394088
Average Silhouette for cluster 1 = 0.256768428122

Average Silhouette for entire dataset = 0.365087812239



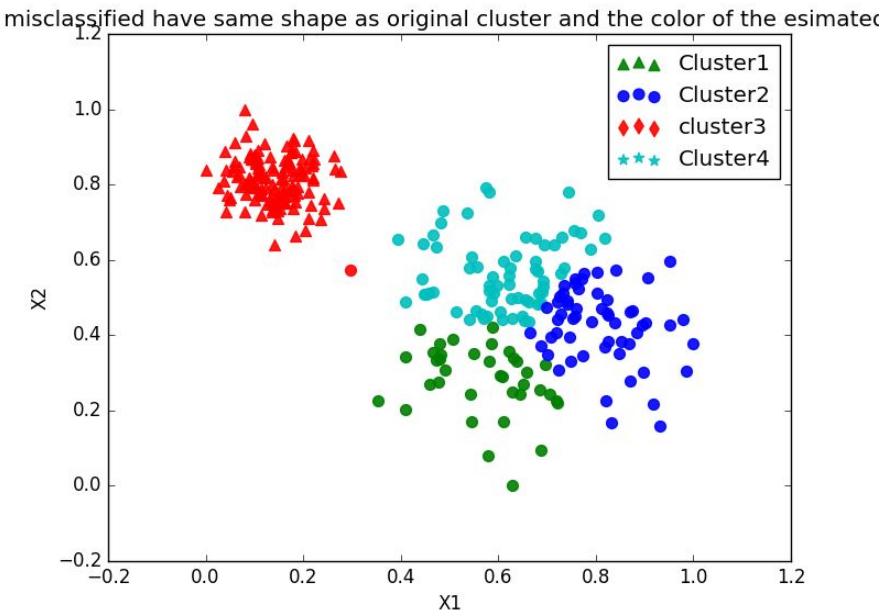
(C.2) On running the K means clustering algorithm for **3 times** on **TwoDimEasy** dataset for **k=4** and **different initial centroids**, the following results were obtained :-

(Trial 1) Initial Centroids :

	0	1	2	3
0	146	0.409995	0.342893	2
1	206	0.576400	0.450343	2
2	52	0.188670	0.778221	1
3	221	0.743960	0.778827	2

CONFUSION MATRIX

	1	2
1	138.0	0.0
2	1.0	67.0



Within cluster SSE of cluster 0 = 0.6510476924448985
Within cluster SSE of cluster 1 = 0.9427427195181899
Within cluster SSE of cluster 2 = 1.0145424355111972
Within cluster SSE of cluster 3 = 1.2431734633500278
Overall cluster SSE = 3.851506310824312
Between-cluster sum of squares(SSB) = 0.11786507096778455
Average Silhouette for cluster 0 = -0.306558342762
Average Silhouette for cluster 1 = -0.386764389711
Average Silhouette for cluster 2 = 0.34074139894
Average Silhouette for cluster 3 = -0.435904704753
Average Silhouette for entire dataset = -0.0507692988715

(Trial 2)

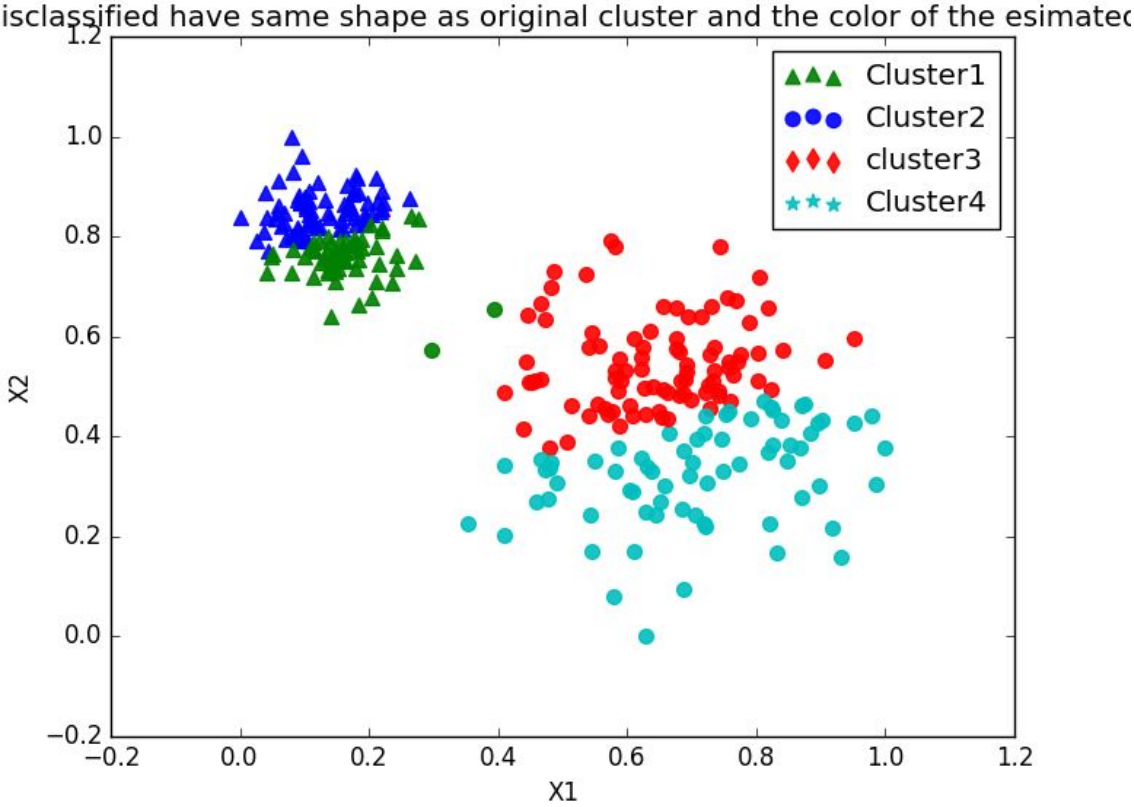
Initial Centroids :

	0	1	2	3
0	106	0.124352	0.748343	1
1	91	0.154995	0.790691	1
2	2	0.197873	0.867769	1
3	180	0.479655	0.378741	2

CONFUSION MATRIX

	1	2
1	77.0	0.0
2	2.0	70.0

misclassified have same shape as original cluster and the color of the esimatec



Within cluster SSE of cluster 0 = 0.3520146432694274
Within cluster SSE of cluster 1 = 0.37058068455595633
Within cluster SSE of cluster 2 = 1.9189557463406588
Within cluster SSE of cluster 3 = 2.380920817654045
Overall cluster SSE = 5.022471891820087
Between-cluster sum of squares(SSB) = 0.23244250051437465
Average Silhouette for cluster 0 = -0.52732540357
Average Silhouette for cluster 1 = -0.539652873811
Average Silhouette for cluster 2 = -0.326359911012
Average Silhouette for cluster 3 = -0.457248996087
Average Silhouette for entire dataset = -0.453848644752

(Trial 3)

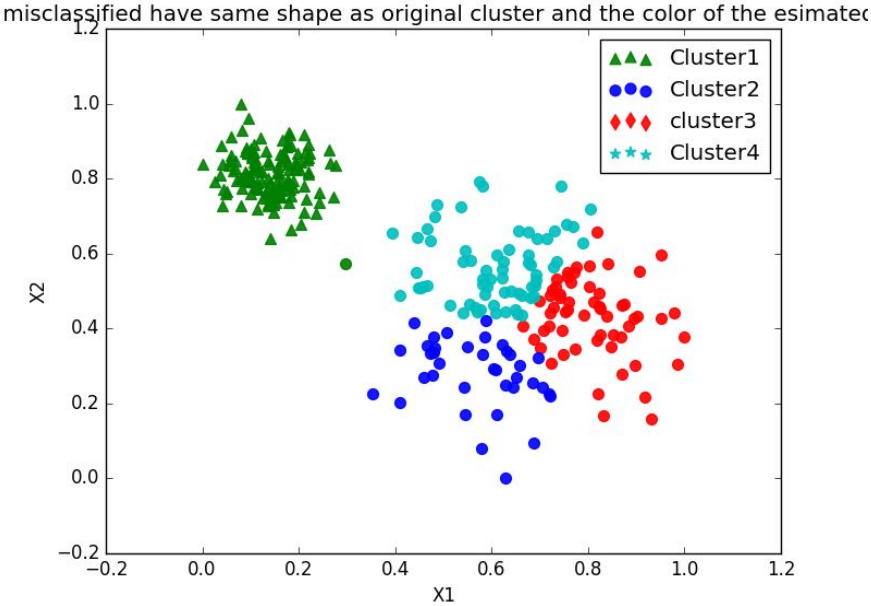
Initial Centroids :

	0	1	2	3
0	53	0.136888	0.726815	1
1	232	0.513306	0.461493	2
2	226	0.691174	0.529691	2
3	289	0.622895	0.559217	2

CONFUSION MATRIX

	1	2
1	138.0	0.0
2	1.0	66.0

misclassified have same shape as original cluster and the color of the esimatec



Within cluster SSE of cluster 0 = 1.0145424355111972
Within cluster SSE of cluster 1 = 0.6510476924448985
Within cluster SSE of cluster 2 = 0.9954856220695025
Within cluster SSE of cluster 3 = 1.1906080590837465
Overall cluster SSE = 3.8516838091093444
Between-cluster sum of squares(SSB) = 0.11786507096778455
Average Silhouette for cluster 0 = 0.337464547529
Average Silhouette for cluster 1 = -0.305221053859
Average Silhouette for cluster 2 = -0.39676016992
Average Silhouette for cluster 3 = -0.427600695004
Average Silhouette for entire dataset = -0.0520644753729

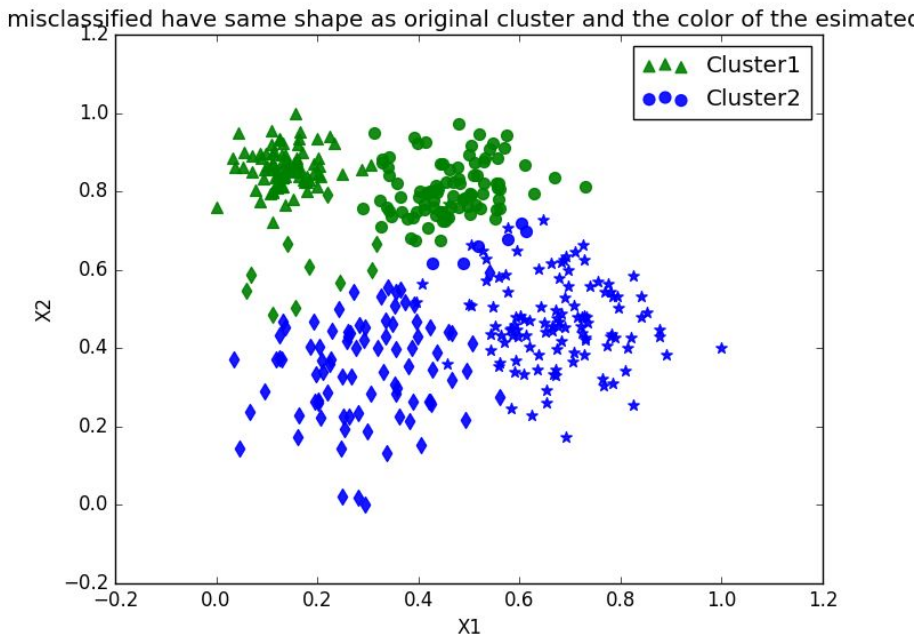
(C.3) On running the K means clustering algorithm for **3 times** on **TwoDimHard** dataset for **k=2** and **different initial centroids**, the following results were obtained :-

(**Trial1**)Initial Centroids :

	0	1	2	3
0	36	0.112577	0.797182	1
1	168	0.489043	0.779873	2

CONFUSION MATRIX

	1	2	3	4
1	0.0	89.0	0.0	0.0
2	0.0	94.0	0.0	6.0
3	0.0	10.0	0.0	87.0
4	0.0	1.0	0.0	113.0



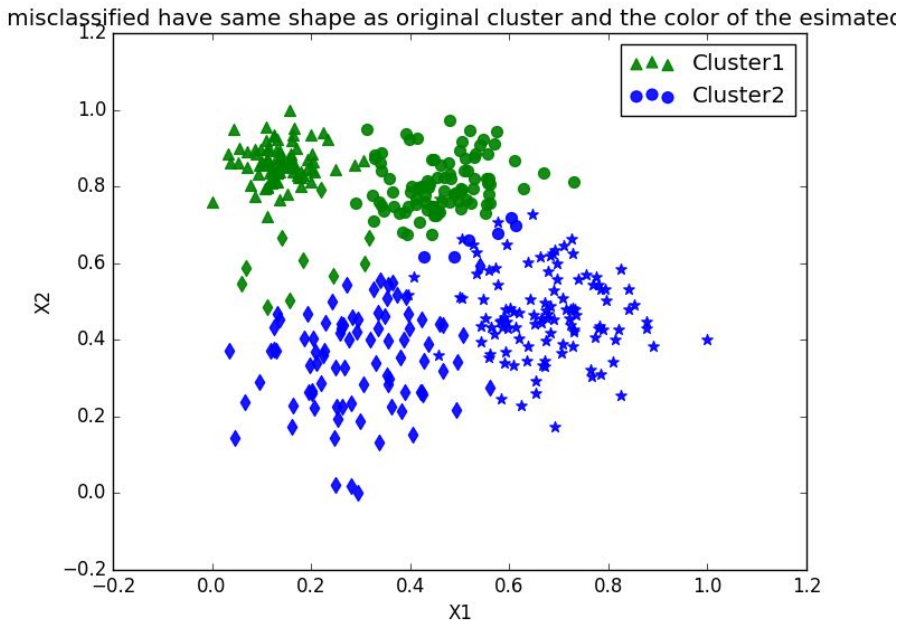
Within cluster SSE of cluster 0 = 7.215588152163626
Within cluster SSE of cluster 1 = 13.110047874668584
Overall cluster SSE = 20.32563602683222
Between-cluster sum of squres(SSB) = 0.05400170758467871
Average Silhouette for cluster 0 = 0.0627875763038
Average Silhouette for cluster 1 = -0.0884861768354
Average Silhouette for entire dataset = -0.0151184065629

(Trial 2) Initial Centroids :

	0	1	2	3
0	37	0.032177	0.885737	1
1	84	0.112260	0.722118	1

CONFUSION MATRIX

	1	2	3	4
1	0.0	89.0	0.0	0.0
2	0.0	94.0	0.0	6.0
3	0.0	10.0	0.0	87.0
4	0.0	1.0	0.0	113.0



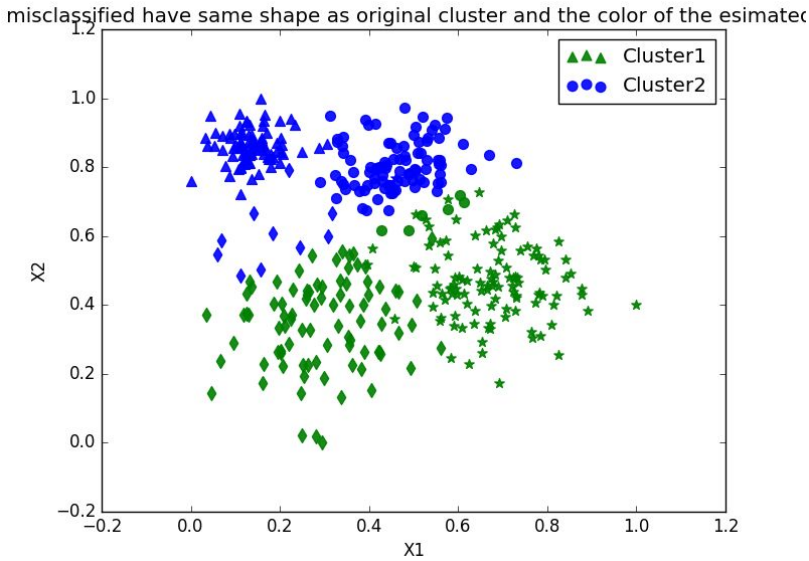
Within cluster SSE of cluster 0 = 7.215588152163626
Within cluster SSE of cluster 1 = 13.110047874668584
Overall cluster SSE = 20.32563602683222
Between-cluster sum of squares(SSB) = 0.05400170758467871
Average Silhouette for cluster 0 = 0.0627875763038
Average Silhouette for cluster 1 = -0.0884861768354
Average Silhouette for entire dataset = -0.0151184065629

(Trial 3) Initial Centroids :

	0	1	2	3
0	107	0.519165	0.660183	2
1	177	0.437720	0.797034	2

CONFUSION MATRIX

	1	2	3	4
1	0.0	89.0	0.0	0.0
2	0.0	94.0	0.0	6.0
3	0.0	10.0	0.0	87.0
4	0.0	1.0	0.0	113.0



Within cluster SSE of cluster 0 = 13.110047874668584
Within cluster SSE of cluster 1 = 7.215588152163626
Overall cluster SSE = 20.325636026832203
Between-cluster sum of squares(SSB) = 0.05400170758467871
Average Silhouette for cluster 0 = -0.0884861768354
Average Silhouette for cluster 1 = 0.0627875763038
Average Silhouette for entire dataset = -0.0151184065629

(C.3) On running the K means clustering algorithm for **3 times** on **TwoDimHard** dataset for **k=4** and **different initial centroids**, the following results were obtained :-

(Trial 1)

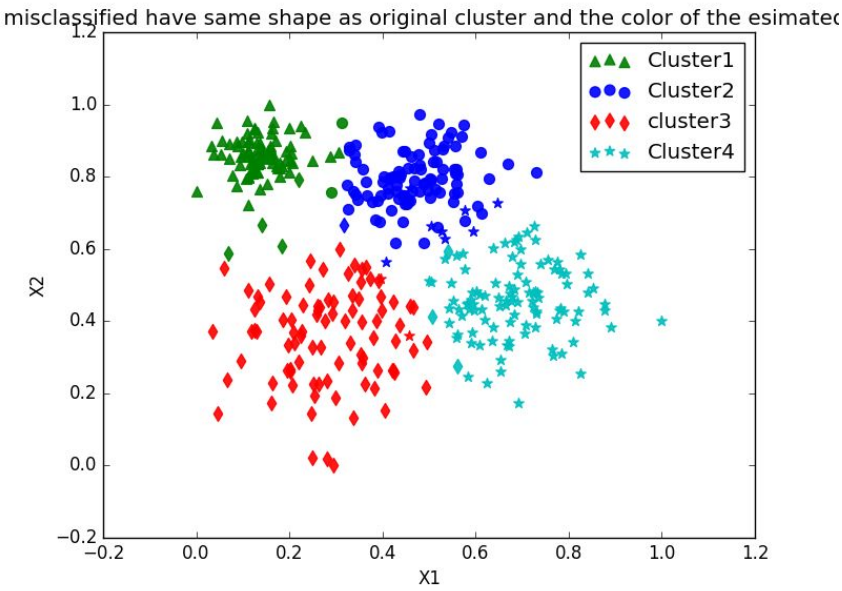
Initial Centroids :

	0	1	2	3
0	228	0.353077	0.509975	3
1	345	0.651153	0.441354	4

2 287 0.609048 0.333088 4
3 351 0.821541 0.426797 4

CONFUSION MATRIX

	1	2	3	4
1	89.0	0.0	0.0	0.0
2	2.0	98.0	0.0	0.0
3	4.0	1.0	89.0	3.0
4	0.0	8.0	2.0	104.0



Within cluster SSE of cluster 0 = 0.6917423729098394
Within cluster SSE of cluster 1 = 1.4358677253052665
Within cluster SSE of cluster 2 = 2.609513314076817
Within cluster SSE of cluster 3 = 2.0419771806324722
Overall cluster SSE = 6.779100592924391
Between-cluster sum of squares(SSB) = 0.126051619389429
Average Silhouette for cluster 0 = 0.0500645554432
Average Silhouette for cluster 1 = -0.153299412059
Average Silhouette for cluster 2 = -0.145303987092
Average Silhouette for cluster 3 = -0.0905880093912
Average Silhouette for entire dataset = -0.0864062103837

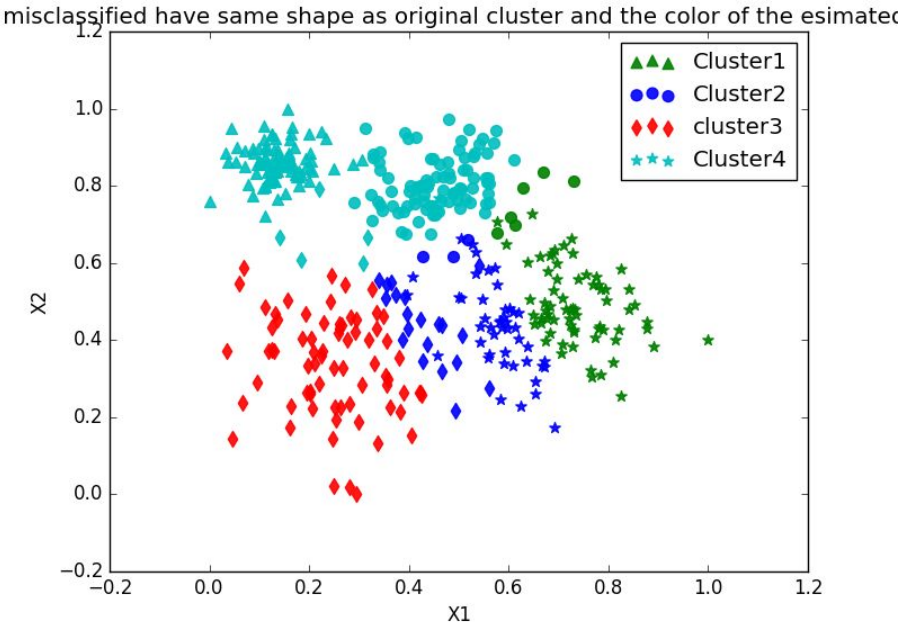
(Trial 2)

Initial Centroids :

	0	1	2	3
0	359	0.724247	0.437831	4
1	345	0.651153	0.441354	4
2	280	0.339932	0.555972	3
3	12	0.193617	0.870642	1

CONFUSION MATRIX

	1	2	3	4
1	0.0	89.0	0.0	0.0
2	0.0	91.0	0.0	3.0
3	0.0	5.0	72.0	20.0
4	0.0	1.0	0.0	45.0



Within cluster SSE of cluster 0 = 1.5079898673927592

Within cluster SSE of cluster 1 = 1.401670196275776

Within cluster SSE of cluster 2 = 1.874756228717752

Within cluster SSE of cluster 3 = 6.1949625527992405

Overall cluster SSE = 10.979378845185538

Between-cluster sum of squares(SSB) = 0.05755953163363772

Average Silhouette for cluster 0 = -0.37643589551

Average Silhouette for cluster 1 = -0.505314476734

Average Silhouette for cluster 2 = -0.253696298599

Average Silhouette for cluster 3 = -0.0512318349187

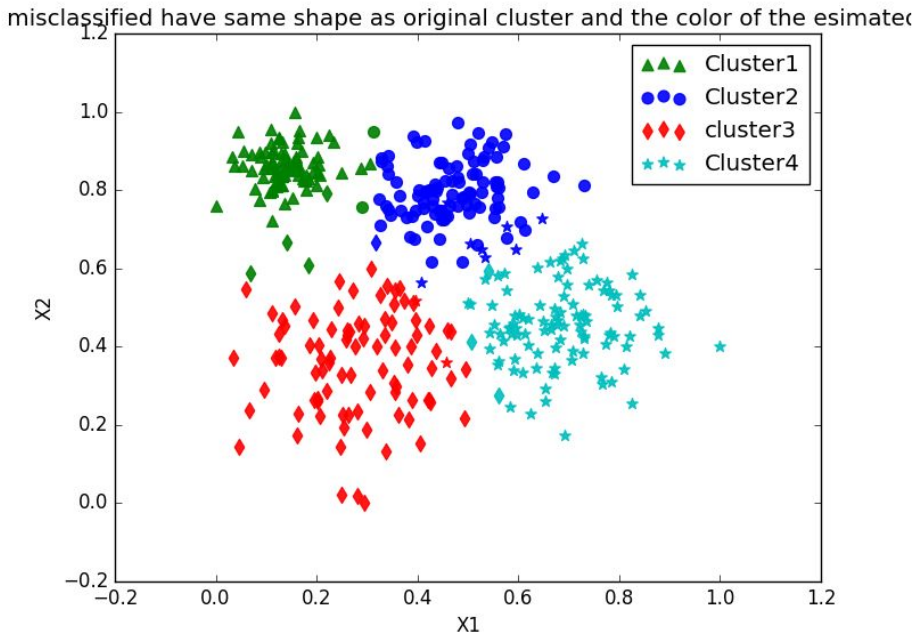
Average Silhouette for entire dataset = -0.225032238699

(Trial3) Initial Centroids :

	0	1	2	3
0	17	0.136234	0.894423	1
1	31	0.153041	0.780761	1
2	369	0.591471	0.479016	4
3	330	0.682019	0.417745	4

CONFUSION MATRIX

	1	2	3	4
1	89.0	0.0	0.0	0.0
2	2.0	98.0	0.0	0.0
3	4.0	1.0	89.0	3.0
4	0.0	8.0	2.0	104.0



Within cluster SSE of cluster 0 = 0.6917423729098394
Within cluster SSE of cluster 1 = 1.4358677253052665
Within cluster SSE of cluster 2 = 2.609513314076817
Within cluster SSE of cluster 3 = 2.0419771806324722
Overall cluster SSE = 6.779100592924391
Between-cluster sum of squares(SSB) = 0.126051619389429
Average Silhouette for cluster 0 = 0.0500645554432
Average Silhouette for cluster 1 = -0.153299412059
Average Silhouette for cluster 2 = -0.145303987092
Average Silhouette for cluster 3 = -0.0905880093912
Average Silhouette for entire dataset = -0.0864062103837

ANALYSIS OF THE RESULTS GOT ABOVE:

Points to keep in mind:-

- TwoDimEasy.csv has 2 true clusters and TwoDimHard.csv has 4 true clusters.
- Always, a smaller Sum of squared error(SSE) indicates good clustering
- Silhouette width values range between -1 and 1.
- Positive silhouette width values indicate that on an average the points of a cluster are closer to other points belonging to the same cluster as opposed to points in neighboring clusters.
- High values of SSB(between cluster sum of square error) is preferred. SSB measures how far each cluster is from the global mean.

Here's how metrics changed as the initial centroids are changed:-

- Initial centroids are chosen randomly. Surprisingly, K means algorithm converged to the same final cluster centroids irrespective of the random initial centroids.
- Because of the convergence mentioned above, the metrics computed always had the same value for a given value of k (num of clusters).

Section D

On running the K means clustering algorithm for **2 times** on **TwoDimEasy** dataset for **k=3** and **different initial centroids**, the following results were obtained :-

(Trial 1) Initial Centroids :

	0	1	2	3
0	90	0.180855	0.772362	1
1	2	0.197873	0.867769	1
2	114	0.173377	0.782720	1

CONFUSION MATRIX

	1	2
1	93.0	0.0
2	1.0	159.0

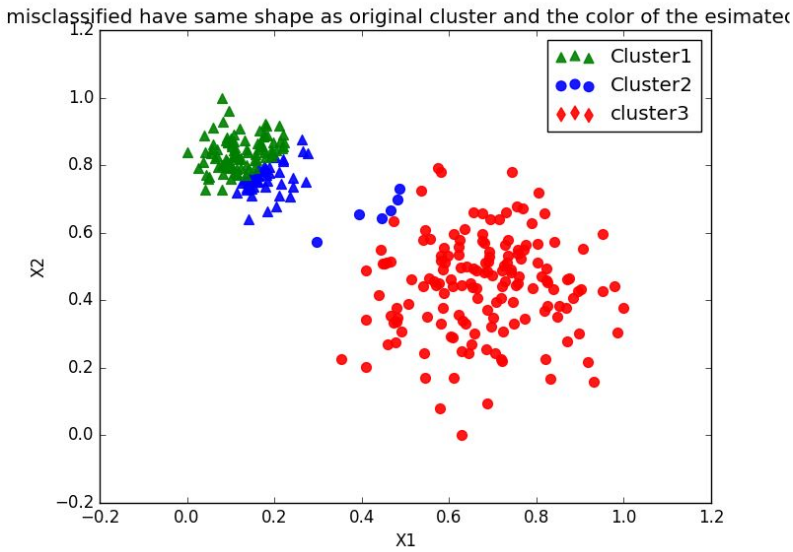
Within cluster SSE of cluster 0 = 6.336358580502975
Within cluster SSE of cluster 1 = 0.3892807959874711
Within cluster SSE of cluster 2 = 0.5207779291118367
Overall cluster SSE = 7.246417305602281
Between-cluster sum of squares(SSB) = 0.22863133757103865
Average Silhouette for cluster 0 = 0.227085832425
Average Silhouette for cluster 1 = -0.571927760009
Average Silhouette for cluster 2 = -0.510375030256
Average Silhouette for entire dataset = -0.12916403403

(Trial 2) Initial Centroids :

	0	1	2	3
0	5	0.089061	0.818822	1
1	82	0.044676	0.772584	1
2	250	0.684703	0.254335	2

CONFUSION MATRIX

	1	2
1	47.0	0.0
2	6.0	156.0



Within cluster SSE of cluster 0 = 0.4649016713398504
Within cluster SSE of cluster 1 = 0.5880764629111866
Within cluster SSE of cluster 2 = 6.04863954806572
Overall cluster SSE = 7.101617682316754
Between-cluster sum of squares(SSB) = 0.0
Average Silhouette for cluster 0 = -0.450172490952

Average Silhouette for cluster 1 = -0.622139563715
Average Silhouette for cluster 2 = 0.139898516052
Average Silhouette for entire dataset = -0.173716416831

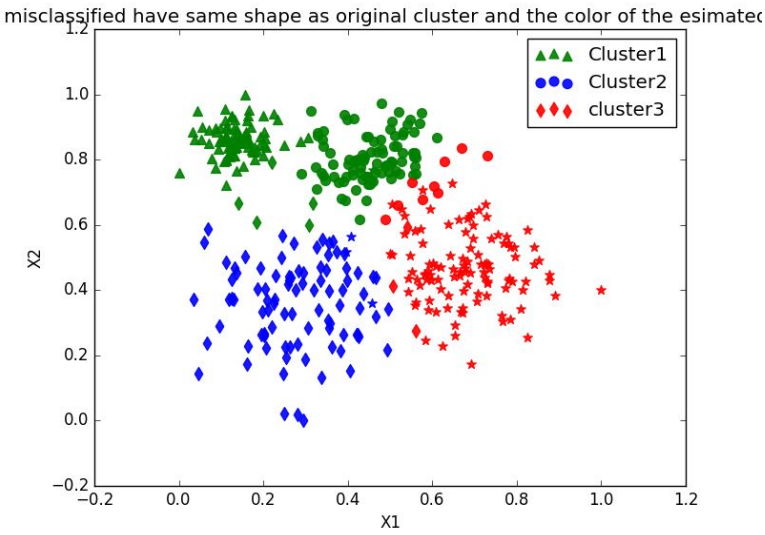
(D2) On running the K means clustering algorithm for **2 times** on **TwoDimHard** dataset for **k=3** and **different initial centroids**, the following results were obtained :-

(Trial 1) Initial Centroids :

	0	1	2	3
0	160	0.342939	0.887942	2
1	276	0.281541	0.234007	3
2	188	0.501880	0.794359	2

CONFUSION MATRIX

	1	2	3	4
1	0.0	89.0	0.0	0.0
2	0.0	91.0	0.0	9.0
3	0.0	5.0	89.0	3.0
4	0.0	1.0	3.0	110.0



Within cluster SSE of cluster 0 = 6.180523188840676
Within cluster SSE of cluster 1 = 2.7072950108383997
Within cluster SSE of cluster 2 = 3.118140794043671
Overall cluster SSE = 12.005958993722748
Between-cluster sum of squares(SSB) = 0.05745472517152942
Average Silhouette for cluster 0 = -0.0274771441458
Average Silhouette for cluster 1 = -0.112065615572

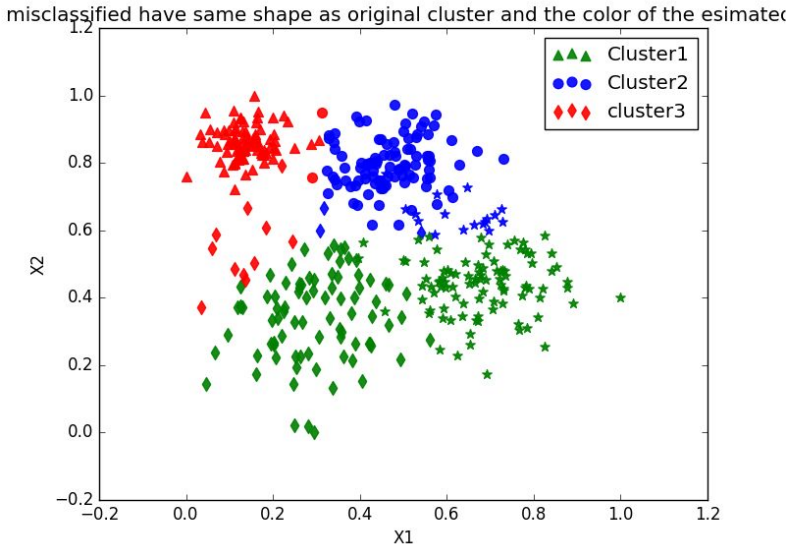
Average Silhouette for cluster 2 = -0.0658098747327
Average Silhouette for entire dataset = -0.0586239754028

(Trial 2) Initial Centroids :

	0	1	2	3
0	178	0.400077	0.785608	2
1	152	0.397814	0.923318	2
2	35	0.100867	0.865725	1

CONFUSION MATRIX

	1	2	3	4
1	89.0	0.0	0.0	0.0
2	2.0	98.0	0.0	0.0
3	11.0	3.0	0.0	83.0
4	0.0	17.0	0.0	97.0



Within cluster SSE of cluster 0 = 10.898602460444785
Within cluster SSE of cluster 1 = 2.1499670237909827
Within cluster SSE of cluster 2 = 1.6208091164460252
Overall cluster SSE = 14.669378600681803
Between-cluster sum of squares(SSB) = 0.1154487234326876
Average Silhouette for cluster 0 = -0.260018319224
Average Silhouette for cluster 1 = -0.158914034624
Average Silhouette for cluster 2 = 0.00968555625949
Average Silhouette for entire dataset = -0.161418067019

Analysis of results obtained for TwoDimEasy dataset:-

As the **k** value increased (2 to 4), the overall SSE tended to drop and the overall SSB increased. I'll choose a higher value of k for TwoDimEasy dataset even though clearly there are only 2 TRUE clusters.

Analysis of results obtained for TwoDimHard dataset:-

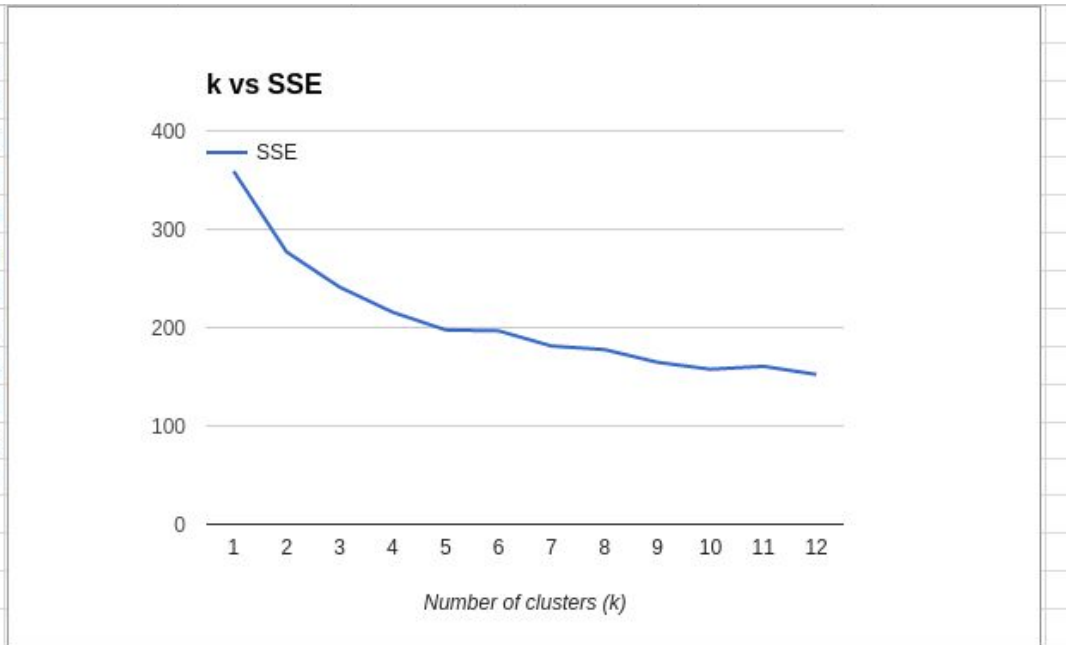
As **k** value increased from (2 to 4), the overall SSE dropped and the overall SSB slightly increased (which is a good sign!!). Interestingly, as k value increased, the intra cluster SSE dropped which indicates the tightness of clusters. I'll choose k=4, for TwoDimHard.csv.

For the Wine dataset:

Sections (E) & (F)

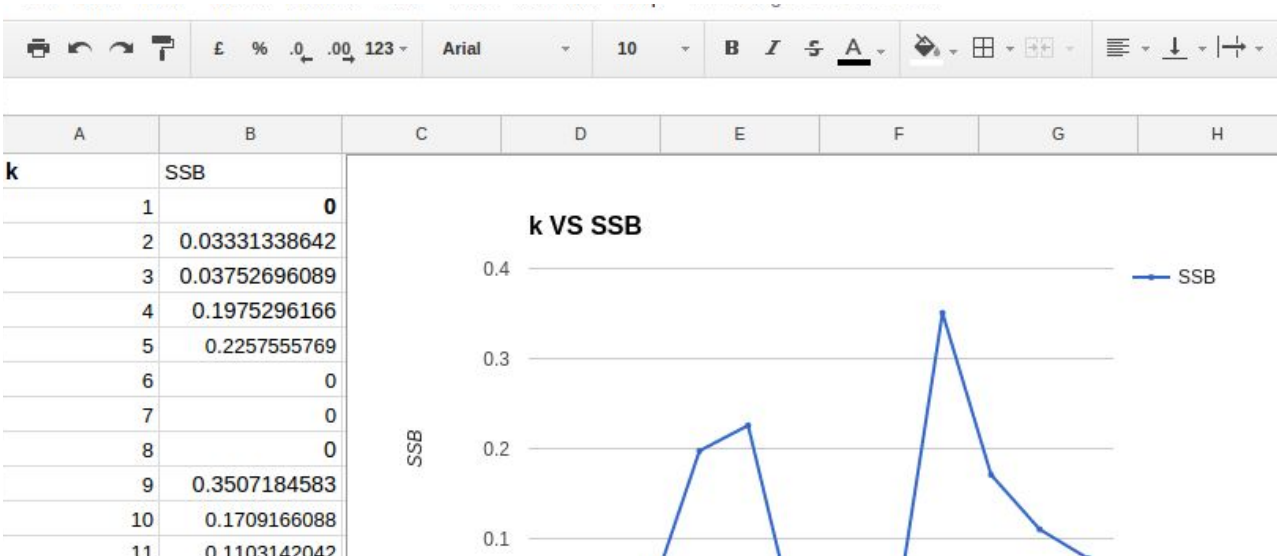
- The wine dataset values are normalized using min_max normalization and then are clustered using k means clustering algorithm for different 'k' values.
- The initial centroids are randomly chosen.
- Here are some visualizations that portray the trend observed: -

k	SSE
1	**359.0599926**
2	**276.9136305**
3	**241.2016398**
4	**215.7014736**
5	**197.6074133**
6	**196.6840994**
7	**181.2504853**
8	**177.6339618**
9	**164.7325882**
10	**157.6143723**
11	160.6051212
12	152.3628689



- From the above table and plot, it's evident that the overall Sum of Squared Error drops as k increases. Overall SSE measures how all data points are distant from its own cluster centroid. A lower SSE is always preferred.

- The following plot is shows the correlation between BSS and the number of clusters. The BSS values increase, become zero and then again spike..



k	SSB
1	0
2	0.03331338642
3	0.03752696089
4	0.1975296166
5	0.2257555769
6	0
7	0
8	0
9	0.3507184583
10	0.1709166088
11	0.1103142042

- From the above visualizations, it's clear that the value **k=9**, produces best cluster. As this clustering produces the highest between sum of square errors and a low sum of squared error.

(Section G): <u>Using the quality attribute as for external validation :-</u>

- Entropy of each of the k clusters formed were estimated. Entropy measures the impurity within a cluster.
- Entropy formula of each cluster is:-

$$H' = \sum_{i=1}^{s} (p_i)(\ln p_i)$$

- Total entropy of the dataset is the weighted sum of the individual entropy values of each cluster.
- Here's a visualization trying to see if there's a trend between the number of clusters (k) and the total entropy value....

- As expected, the clusters became purer for an increase in ‘k’, and then, the entropy value again spiked up.. This can be attributed to overfitting!
- It is not very intuitive to come up with a good value of k, from the above visualization.. But, the previous choice of k=9, is reasonable here too. As it has a low total entropy value.

Part 2 OFF THE SHELF KNN:

- Scikit learn cluster’s k_means was used as the off the shelf clustering tool.
- The **KMeans** algorithm clusters data by trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares. This algorithm requires the number of clusters to be specified. It scales well to large number of samples and has been used across a large range of application areas in many different fields.
- The K-means algorithm aims to choose centroids that minimise the *inertia*, or within-cluster sum of squared criterion:

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (||x_j - \mu_i||^2)$$

- Unnormalized data was used for fitting.
- Parameters used in the following function call are : -

centroids,labels,inertia=sklearn.cluster.k_means(data, n_clusters=k, init='k-means++', precompute_distances='auto', n_init=10, max_iter=300, verbose=False, tol=0.0001, random_state=None, copy_x=True, n_jobs=1,return_n_iter=False)

1. *n_clusters - The number of clusters to form as well as the number of centroids to generate.*
2. *Init - Method for initialization, default to ‘k-means++’: ‘k-means++’ : selects initial cluster centers for k-mean clustering in a smart way to speed up convergence.*
3. *Precompute_distances - Precompute distances (faster but takes more memory).*
4. *‘auto’ : do not precompute distances if n_samples * n_clusters > 12 million. This corresponds to about 100MB overhead per job using double precision.*
5. *N_init - Number of time the k-means algorithm will be run with different centroid seeds. The final results will be the best output of n_init consecutive runs in terms of inertia.*
6. *Max_iter : Maximum number of iterations of the k-means algorithm to run.*

7. *N_jobs :The number of jobs to use for the computation. This works by computing each of the n_init runs in parallel.*

Return values:

***centroid** : float ndarray with shape (k, n_features)*

Centroids found at the last iteration of k-means.

***label** : integer ndarray with shape (n_samples,)*

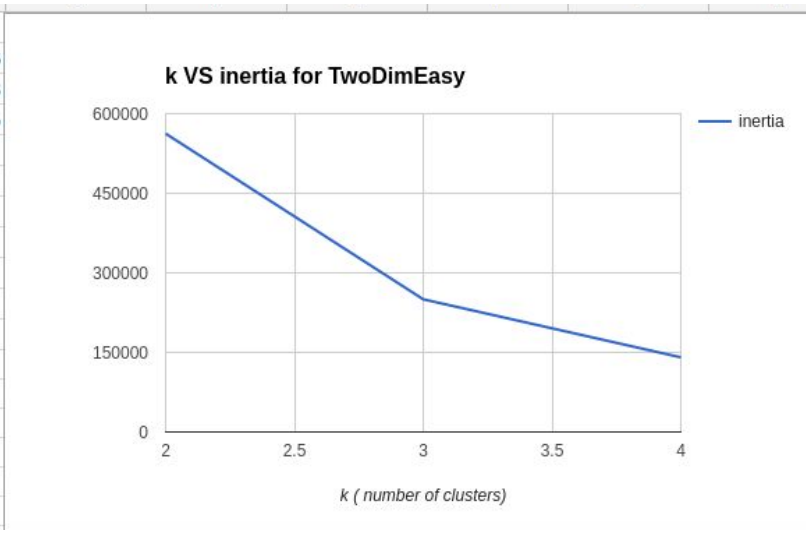
label[i] is the code or index of the centroid the i'th observation is closest to.

***inertia** : float*

The final value of the inertia criterion (sum of squared distances to the closest centroid for all observations in the training set).

Results obtained using this off_the_shelf classifer:-

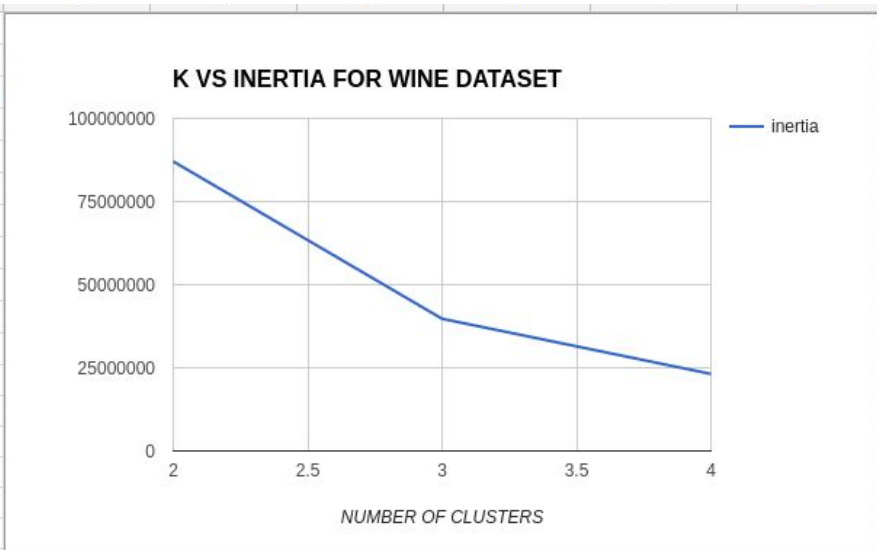
k	inertia
2	**562517.8416**
3	250091.1468
4	140678.815



As expected, the inertia values decrease as k increases for both the TwoDimEasy and TwoDimHard datasets. This trend matches with the metrics established previously. So, k=4 would be a good choice for both TwoDimEasy and TwoDimHard datasets.

For the wine dataset, here are the inertia values obtained :-

k	inertia
2	**87057434.17**
3	39746194.18
4	23161029.14



Thus, kmeans clustering was done.