UNSUPERVISED LEARNING

Francis Mendoza

ASUID: 1213055998

CSE575- Statistical Machine Learning

fmendoz7@asu.edu

1. INTRODUCTION

a. I extracted a 2-D dataset as a .mat file and parsed the relevant data as a list of lists in my program. In this project I used two variations of K-Means Clustering, each initialized twice, to classify the points within this dataset. The entire file was first developed on the Jupyter Notebook environment and then converted into a .py file using the following command

i.
```
jupyter nbconvert --to script Mendoza-CSE575_Project2.ipynb
```

2. STRATEGY #1: Random Initialization

i. Strategy #1 involved randomly configuring the centroids for each individual run of the algorithm in the program main method, until we have reached k desired clusters. The program iterated several times over each individual run until convergence for k specified clusters was reached, until we ran K Means Clustering again, but for k+1 clusters instead.

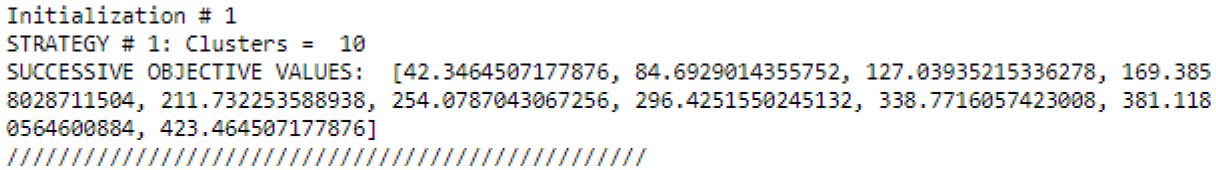
ii. As per my discourse with TA Yuzhen Ding during her Zoom office hours, 3/27, the valid way to randomly initialize points was to select the first k points for k desired clusters for that run.

iii. In the graphs, the x values represented k number of clusters at a particular run, while the y values represented the SUM of the results from the objective function until k clusters were reached

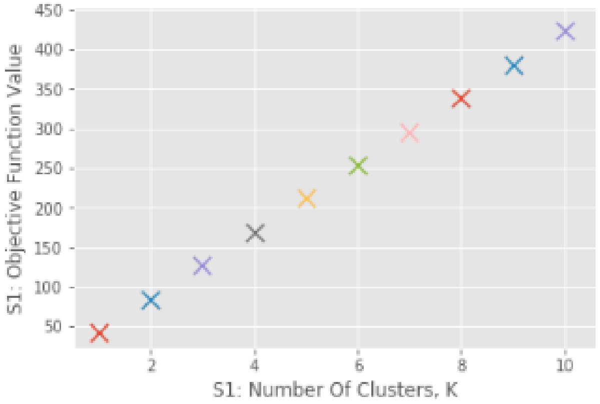
1. For example, if k = 3, the y-value reflected the sum of the objective function values for all k’s until k = 3
2. The objective function is

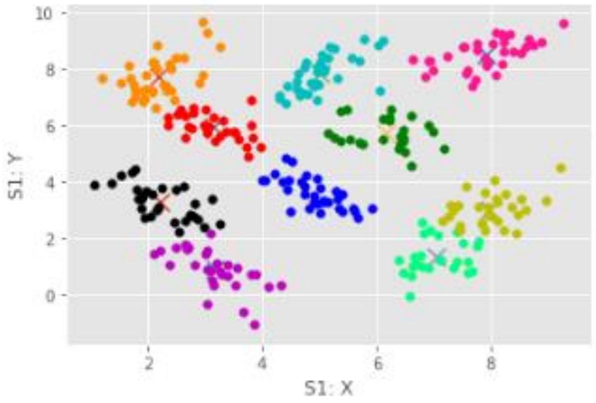
$$\sum_{i=1}^{k} \sum_{\mathbf{x} \in D_i} ||\mathbf{x} - \boldsymbol{\mu}_i||^2$$

iv. The results for Initialization #1, Strategy #1, Yuzhen’s Method were:

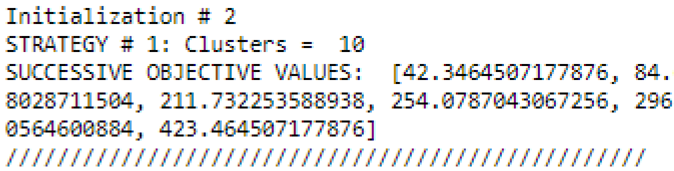


```
Initialization # 1
STRATEGY # 1: Clusters =  10
SUCCESSIVE OBJECTIVE VALUES:  [42.3464507177876, 84.6929014355752, 127.03935215336278, 169.385
8028711504, 211.732253588938, 254.0787043067256, 296.4251550245132, 338.7716057423008, 381.118
0564600884, 423.464507177876]
//////////////////////////////////////////////////
```

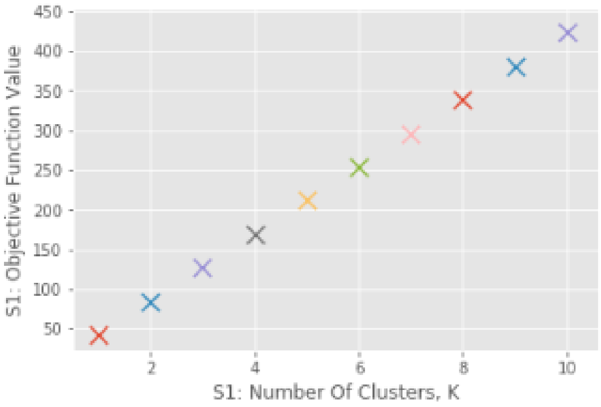


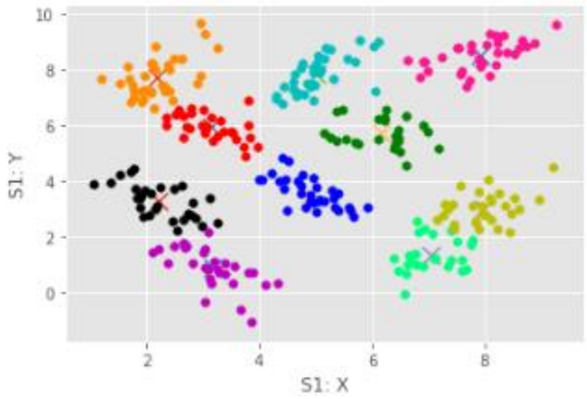


v. The results for Initialization #2, Strategy #1, Yuzhen’s Method, were



```
Initialization # 2
STRATEGY # 1: Clusters =  10
SUCCESSIVE OBJECTIVE VALUES:  [42.3464507177876, 84.6929014355752, 127.03935215336278, 169.385
8028711504, 211.732253588938, 254.0787043067256, 296.4251550245132, 338.7716057423008, 381.118
0564600884, 423.464507177876]
////////////////////////////////////////////////////////
```

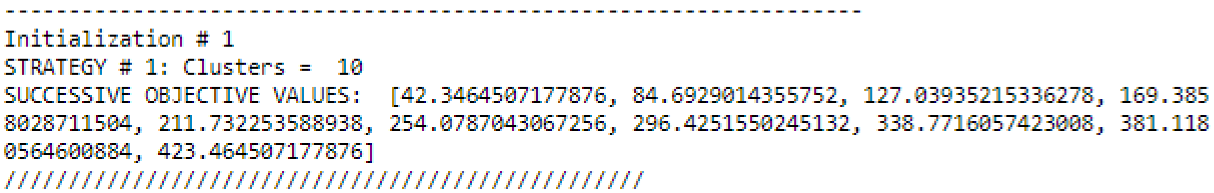




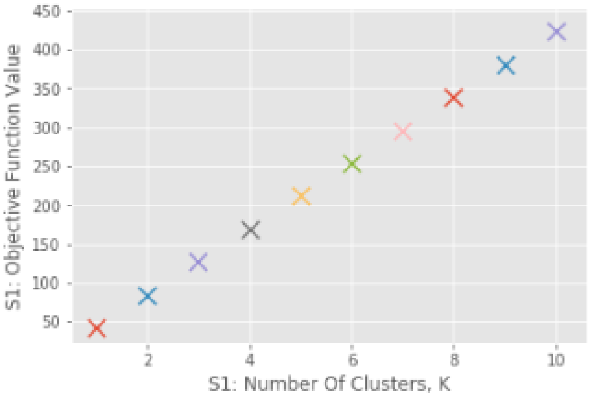
vi. Because we consistently picked the k first coordinates from the dataset as our valid way to randomly initialize (approved by Yuzhen Ding during her TA office hours on 3/27), there was consistency for the initializations

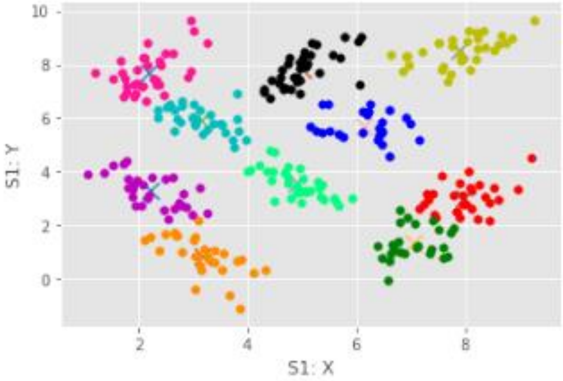
vii. However, K Means Clustering is very sensitive to the initial centroids. Suspecting a possible counter, I developed a secondary random initialization code block that instead relies on the Python “random” library, selecting random coordinates from the dataset using random.choice(dataset_name). The results for this alternative method are:

viii. The results for Initialization #1, Strategy #1, Randomness Library method

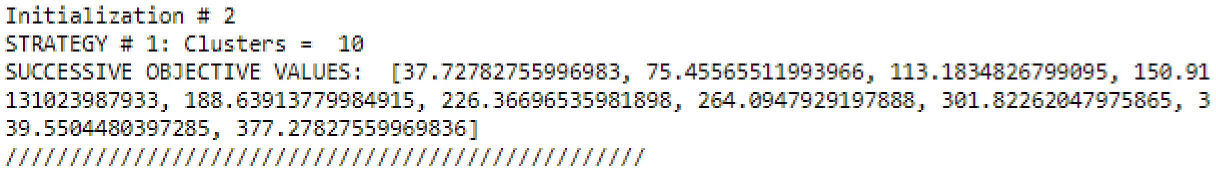


```
-------------------------------------------------------------------
Initialization # 1
STRATEGY # 1: Clusters =  10
SUCCESSIVE OBJECTIVE VALUES:  [42.3464507177876, 84.6929014355752, 127.03935215336278, 169.385
8028711504, 211.732253588938, 254.0787043067256, 296.4251550245132, 338.7716057423008, 381.118
0564600884, 423.464507177876]
//////////////////////////////////////////////////
```

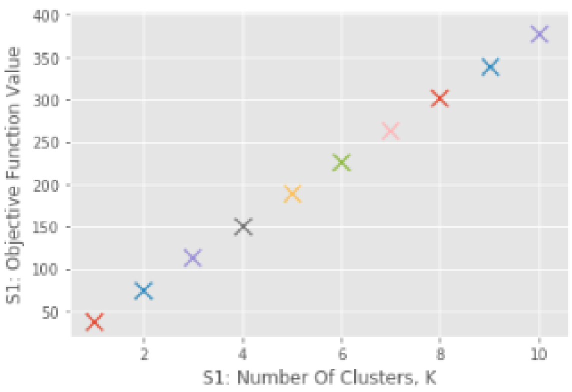


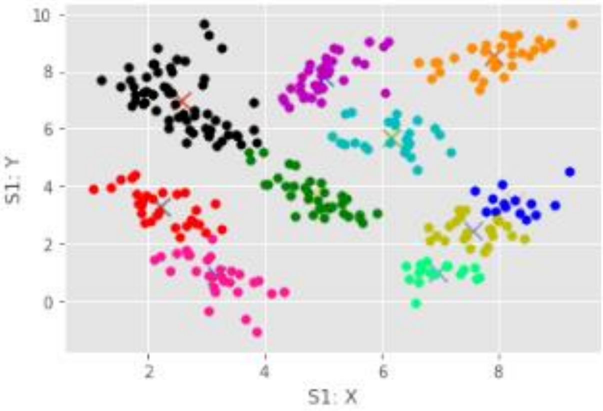


ix. The results for Initialization #2, Strategy #1, Randomness Library method:



```
Initialization # 2
STRATEGY # 1: Clusters =  10
SUCCESSIVE OBJECTIVE VALUES:  [37.72782755996983, 75.45565511993966, 113.1834826799095, 150.91
131023987933, 188.63913779984915, 226.36696535981898, 264.0947929197888, 301.82262047975865, 3
39.5504480397285, 377.27827559969836]
//////////////////////////////////////////////////
```





3. STRATEGY #2: Maximum Average Centroids

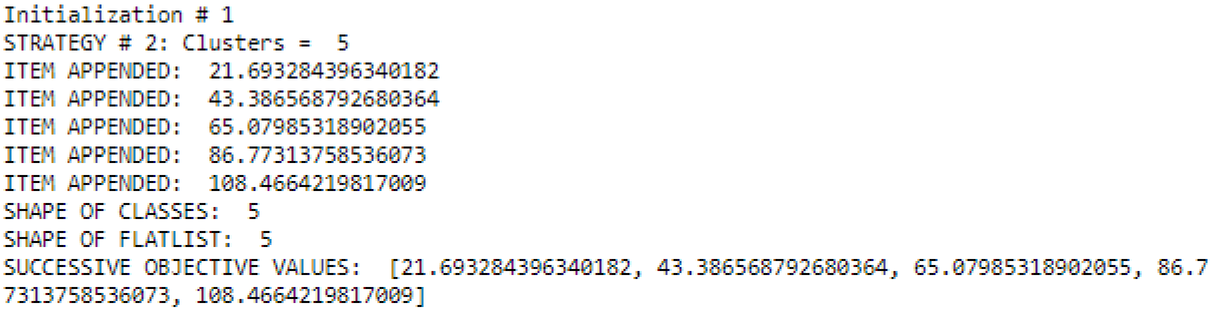
i. Strategy #2 involved having the first centroid being randomly assigned, with subsequent centroids being the point with the maximum average distance in comparison to all other points. The program iterated several times over each individual run until convergence for k specified clusters was reached, until we ran K Means Clustering again, but for k+1 clusters instead.

ii. As per my discourse with TA Yuzhen Ding during her Zoom office hours, 3/27, the valid way to randomly initialize points was to select the first point within the dataset

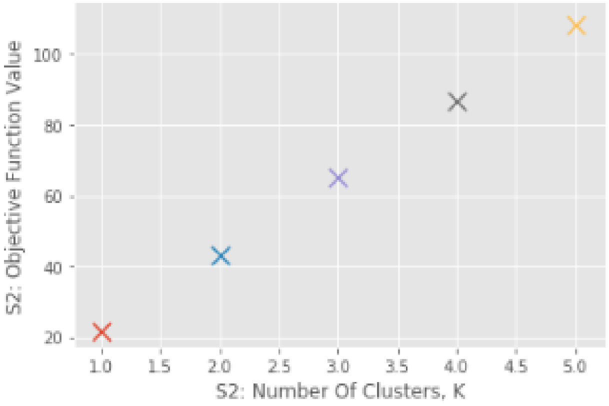
iii. The graphs used the same format as strategy #1. The objective function is

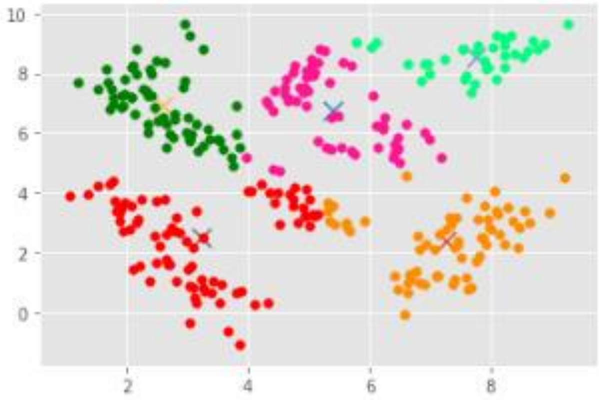
$$\sum_{i=1}^{k} \sum_{\mathbf{x} \in D_i} ||\mathbf{x} - \boldsymbol{\mu}_i||^2$$

iv. The results for Initialization #1, Strategy #2, Yuzhen's Method:

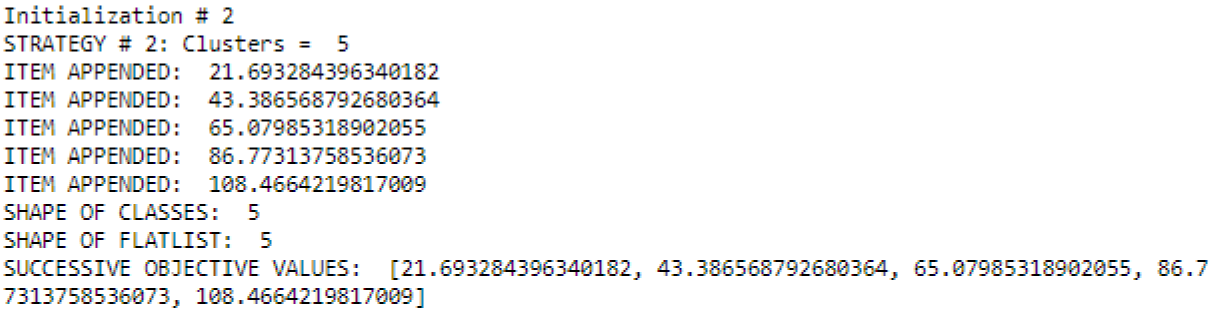


```
Initialization # 1
STRATEGY # 2: Clusters =  5
ITEM APPENDED:  21.693284396340182
ITEM APPENDED:  43.386568792680364
ITEM APPENDED:  65.07985318902055
ITEM APPENDED:  86.77313758536073
ITEM APPENDED:  108.4664219817009
SHAPE OF CLASSES:  5
SHAPE OF FLATLIST:  5
SUCCESSIVE OBJECTIVE VALUES:  [21.693284396340182, 43.386568792680364, 65.07985318902055, 86.7
7313758536073, 108.4664219817009]
```

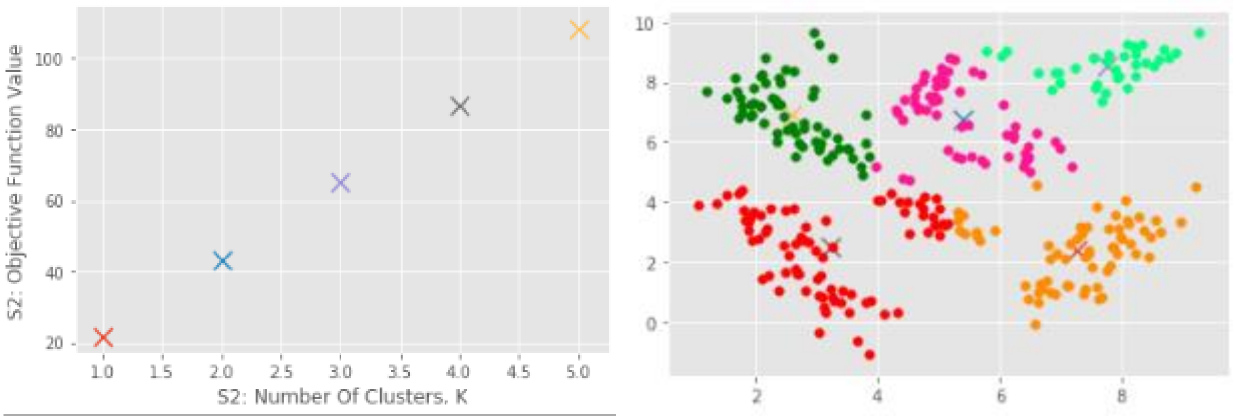




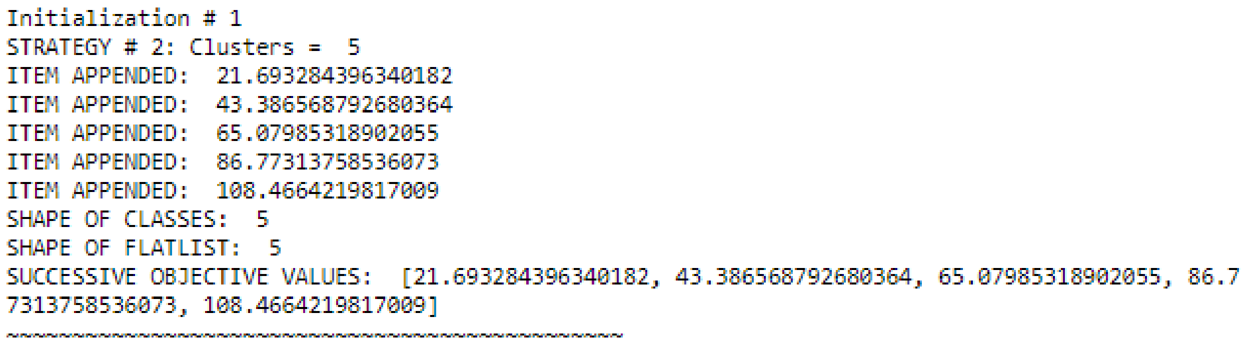
v. The results for Initialization #2, Strategy #2, Yuzhen's Method:



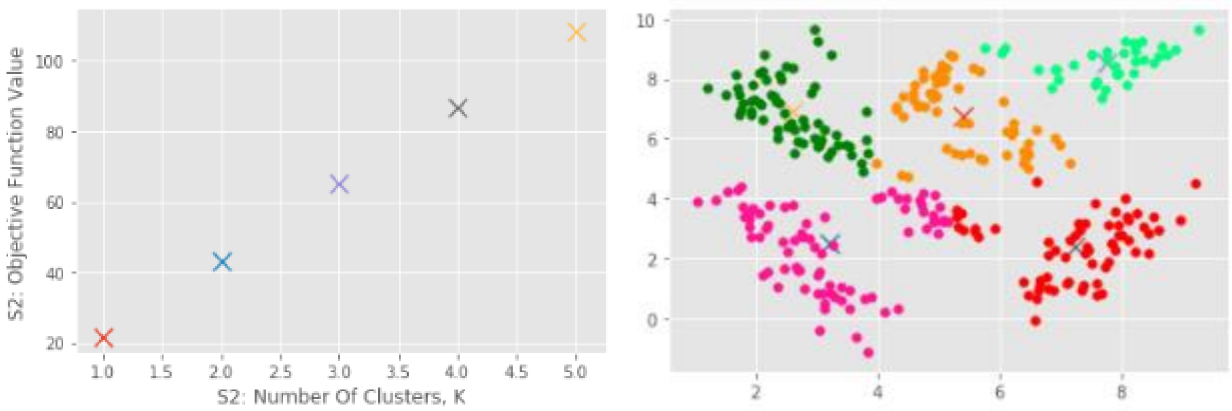
```
Initialization # 2
STRATEGY # 2: Clusters =  5
ITEM APPENDED:  21.693284396340182
ITEM APPENDED:  43.386568792680364
ITEM APPENDED:  65.07985318902055
ITEM APPENDED:  86.77313758536073
ITEM APPENDED:  108.4664219817009
SHAPE OF CLASSES:  5
SHAPE OF FLATLIST:  5
SUCCESSIVE OBJECTIVE VALUES:  [21.693284396340182, 43.386568792680364, 65.07985318902055, 86.7
7313758536073, 108.4664219817009]
```



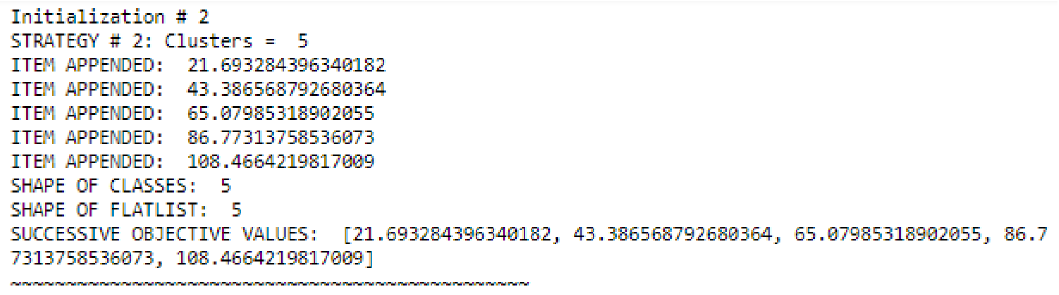
vi. The results for Initialization #1, Strategy #2, Randomness Library method:



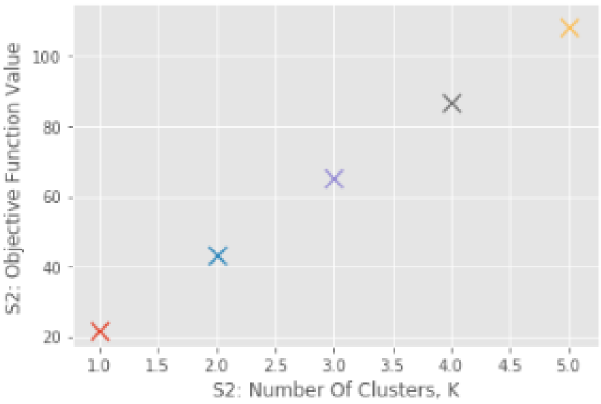
```
Initialization # 1
STRATEGY # 2: Clusters =  5
ITEM APPENDED:  21.693284396340182
ITEM APPENDED:  43.386568792680364
ITEM APPENDED:  65.07985318902055
ITEM APPENDED:  86.77313758536073
ITEM APPENDED:  108.4664219817009
SHAPE OF CLASSES:  5
SHAPE OF FLATLIST:  5
SUCCESSIVE OBJECTIVE VALUES:  [21.693284396340182, 43.386568792680364, 65.07985318902055, 86.7
7313758536073, 108.4664219817009]
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

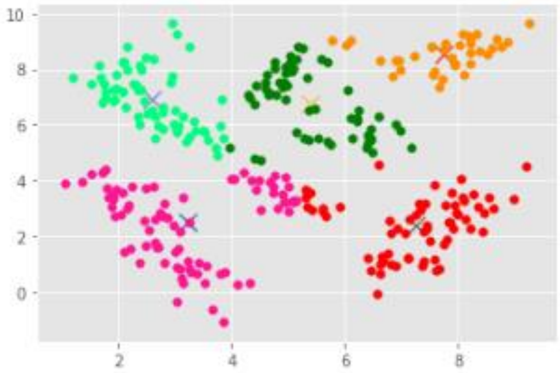


vii. The results for Initialization #1, Strategy #2, Randomness Library method:



```
Initialization # 2
STRATEGY # 2: Clusters =  5
ITEM APPENDED:  21.693284396340182
ITEM APPENDED:  43.386568792680364
ITEM APPENDED:  65.07985318902055
ITEM APPENDED:  86.77313758536073
ITEM APPENDED:  108.4664219817009
SHAPE OF CLASSES:  5
SHAPE OF FLATLIST:  5
SUCCESSIVE OBJECTIVE VALUES:  [21.693284396340182, 43.386568792680364, 65.07985318902055, 86.7
7313758536073, 108.4664219817009]
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```





4. RESULTS

a. Yuzhen's method of picking k centroids for randomly initializing (either all centroids for Strategy #1 or the first centroid for Strategy #2) did not yield any varying results

b. Using the randomness library to initialize centroids did yield different results for multiple runs (initializations)

c. The objective function graph for both strategies #1 and #2, with both Yuzhen's method and the randomness library for initialization, yielded a consistent linear growth. However, for strategies #1, it yielded different values when it utilized the randomness library for initialization. This behavior has not been observed until k = 5 in strategy #2, however

d. Interestingly, for Strategy #2, K Means Clustering operates fine until the 6th cluster. I have emailed the TA's at roughly 15:00 hrs MST and am awaiting a response, but in case I haven't, although the graphs are not listed for the remaining 5 clusters, the code demonstrates the core concept.

e. The core code demonstrates clustering until k = 10 for Strategy #1 and Strategy #2. If you would like to see the progression of clustering for each increasing run for k, please run the code and comment out either Yuzhen's method or the randomness library initialization method for your centroids

f. Although I was unable to fully debug the Strategy #2 code to work for 10 clusters (only operates until k = 6), the core code fully demonstrates the respective operations for both methods, displaying both the objective function graph and the scatterplot, with the objective function having accurate values, as evidenced by this email correspondence with TA Yuzhen Ding

CSE575- Are these reasonable values for objective?

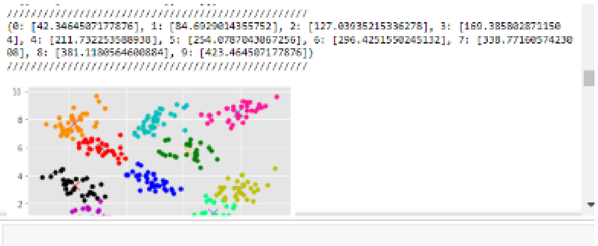
Francis Mendoza <francissamuelmendoza7@gmail.com>
to Yuzhen

Fri, Mar 27, 12:38 PM (1 day ago)

ASUID: 1213055998

Good Afternoon Yuzhen,

I have printed a python dictionary containing the summed distance metrics for each successive k (ie: k2 has k1 and k2, etc.). Are these y values within a reasonable range?



```
////////////////////////////////////////////////////
{0: [42.3464507177876], 1: [84.6920014355752], 2: [127.03935215336278], 3: [169.385802871150
4], 4: [211.732253588938], 5: [254.0787043067256], 6: [296.4251550245132], 7: [338.77160574230
08], 8: [381.1180564600884], 9: [423.464507177876]}
////////////////////////////////////////////////////
```

Very Respectfully,
Francis

Francis Mendoza <francissamuelmendoza7@gmail.com>
to kevin

Fri, Mar 27, 12:55 PM (1 day ago)

•••

Yuzhen Ding
to me

Fri, Mar 27, 1:08 PM (1 day ago)

Looks good to me.

•••