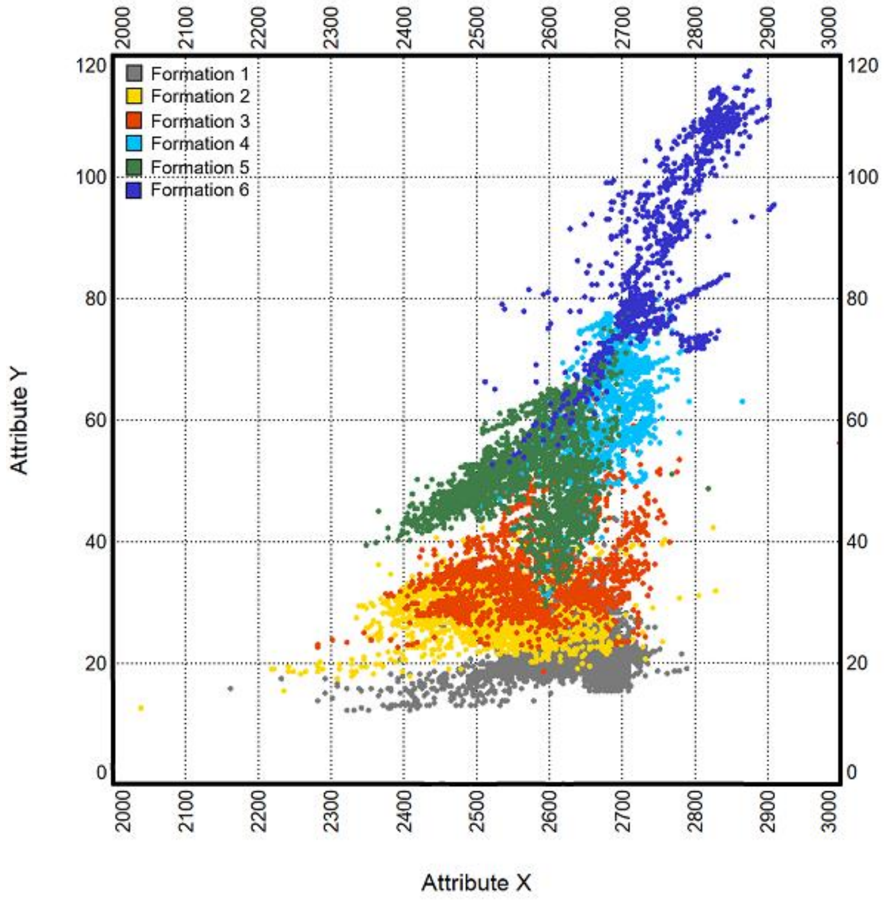
Clustering Challenge for Sound QI

Analyzed by Homayoun Gerami

21 July 2021

The provided cross-plot for guidance, from a similar geological units, and my quick observations:



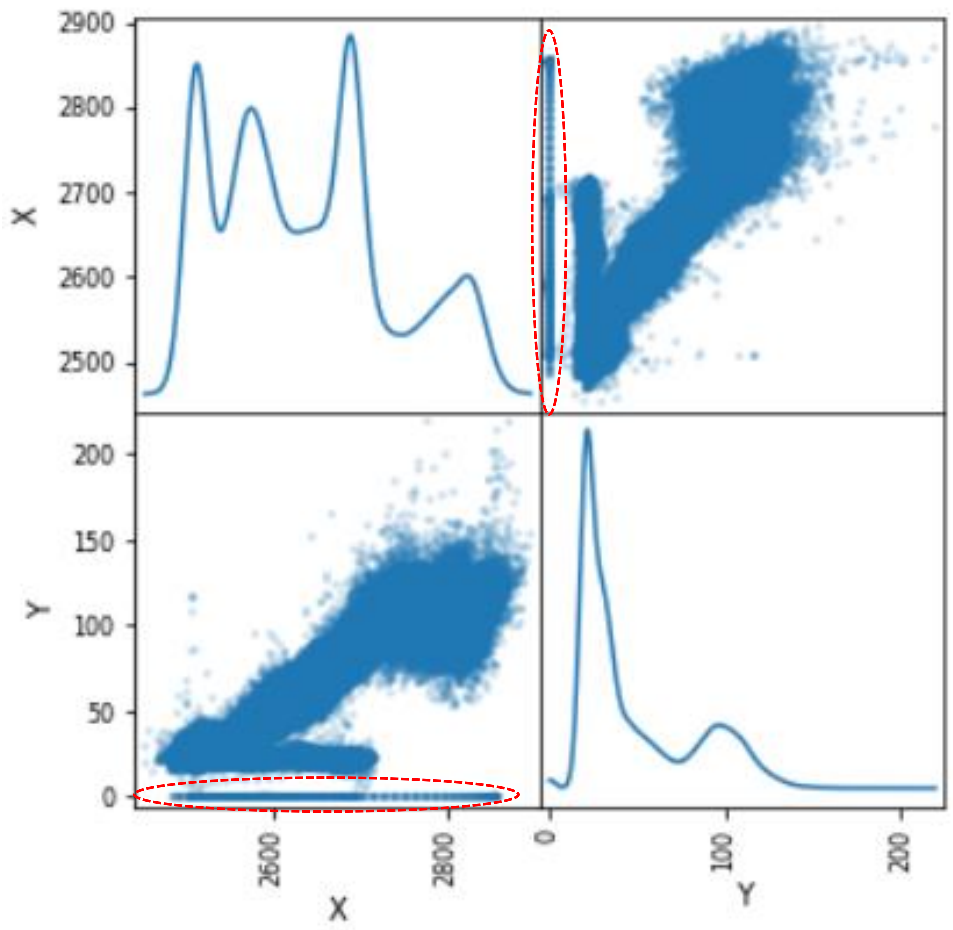
Observations:

- The Attributes “X” and “Y” have very different ranges of values, and hence clustering without a proper scaling maybe dominated by the attribute with larger values

- The overlain classes, suggest better separability of the data points with the attribute “X” than “Y”

Data loading and quick EDA:

Scatter Matrix




```
In [5]: df.sort_values('Y')
```

Out[5]:

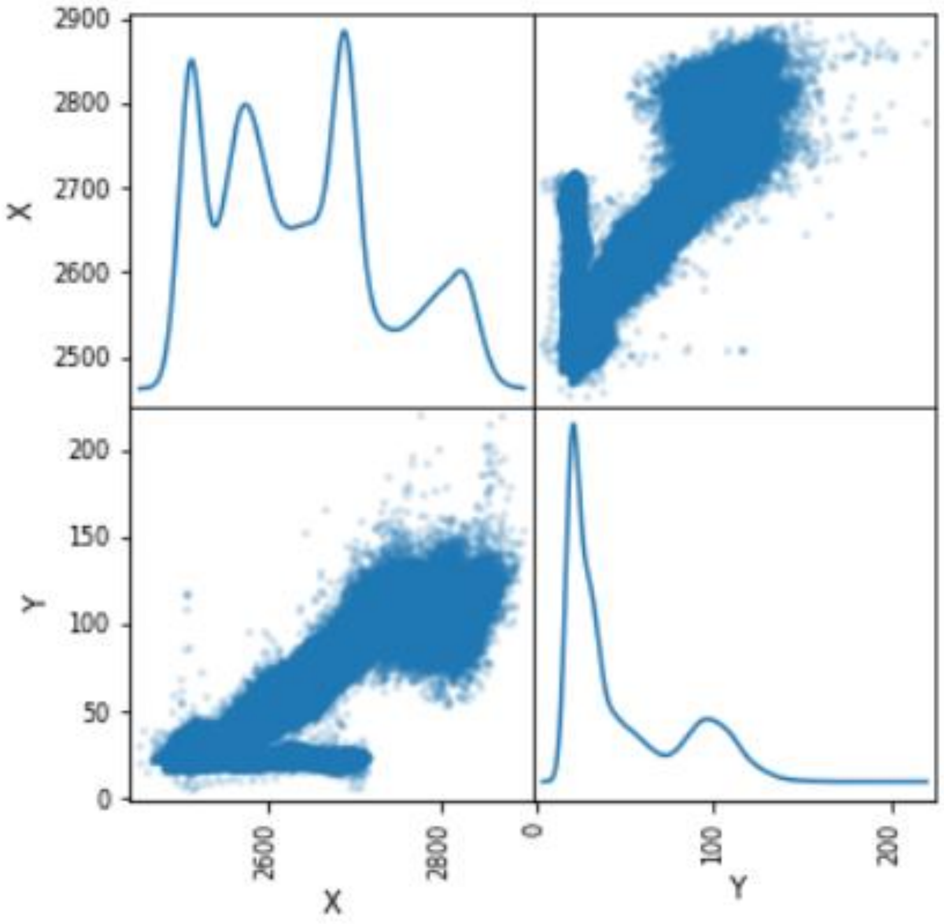
	X	Y
83563	2519.675	0.000000
83331	2678.633	0.000000
83330	2683.430	0.000000
83329	2687.396	0.000000
83328	2690.536	0.000000
...	...	...
43215	2852.009	201.348938
43524	2857.823	201.906586
84325	2856.452	213.977554
43930	2870.404	218.799652
82297	2776.725	219.351196

142410 rows × 2 columns

- I noticed samples with 0.0 values, highlighted in dashed red line, for the Y attribute.
- I have removed them so that clustering have better chance of success

Data loading and quick EDA:

Scatter Matrix after removing 0.0 values



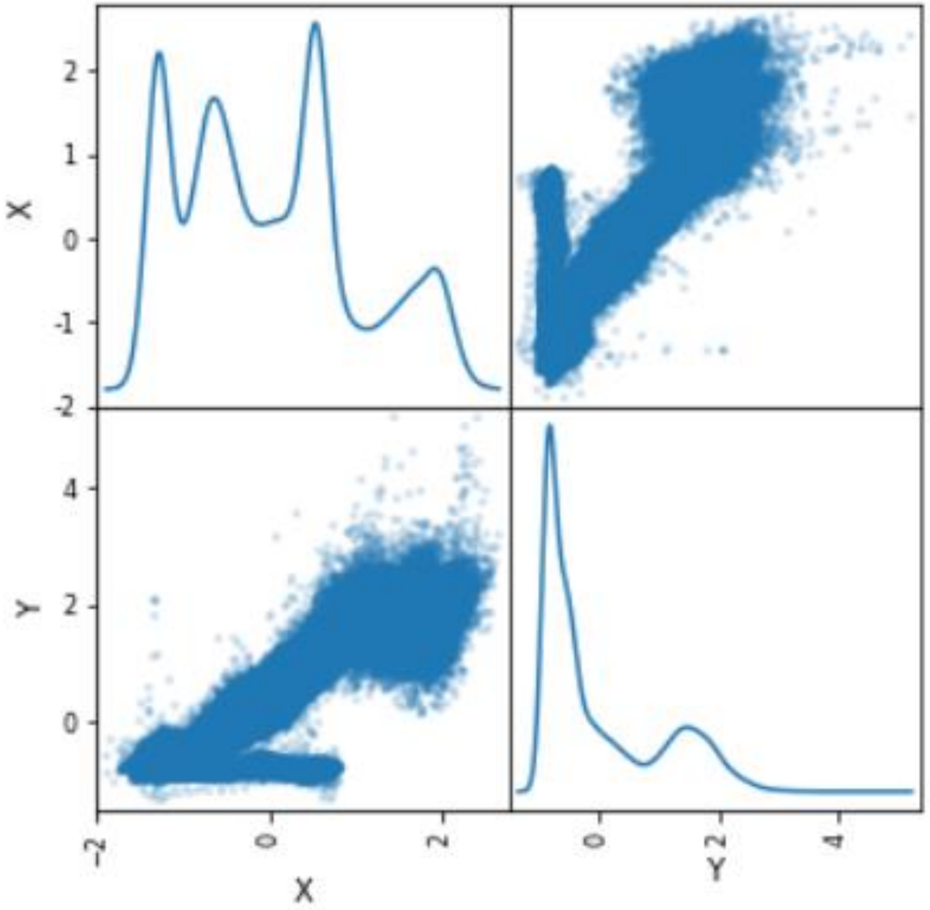
In [134]:
```
df_Cleaned1
```
Out[134]:

	X	Y
0	2690.201	22.937439
1	2679.136	22.541031
2	2663.628	20.859741
3	2652.534	20.203293
4	2647.038	20.485809
...	...	...
142405	2773.997	106.855255
142406	2781.634	112.347260
142407	2793.332	117.831955
142408	2807.608	115.843094
142409	2817.894	106.925797

141501 rows × 2 columns

Applying Standard Scaler on “X” and “Y”

Scatter Matrix after removing 0.0 values & applying Standard-Scaler



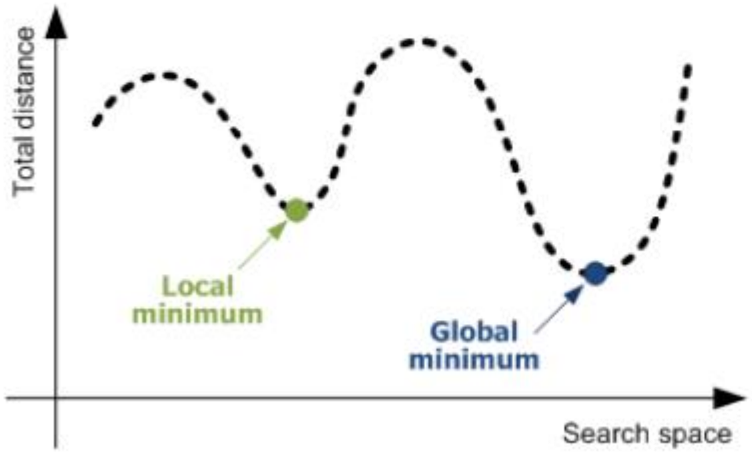
- Standard-Scaler: standardize features by removing the mean and scaling to unit variance [1]; applying this function on attributes is recommended for most of the clustering methods, to work properly and being less biased by original magnitude of the attributes

Clustering methods tested for this challenge:

- K-means
- Gaussian Mixture Model (GMM)
- Agglomerative clustering (AC)
- Spectral Clustering (the result of this method is not presented in this file, as it was not promising)

K-means

- K-means clustering: It is very simple to implement and fast to run. The number of clusters need to be set before clustering, and the algorithm attempt to minimize sum-of-squared distances from each data point to its respective cluster center
- The algorithm always converges, but the results are sensitive to the initial cluster assignments
- For ‘m’ data points together, there are k^m possibilities to converge , K= number of clusters, and hence most of the times the algorithm will converge to a local minimum [2]

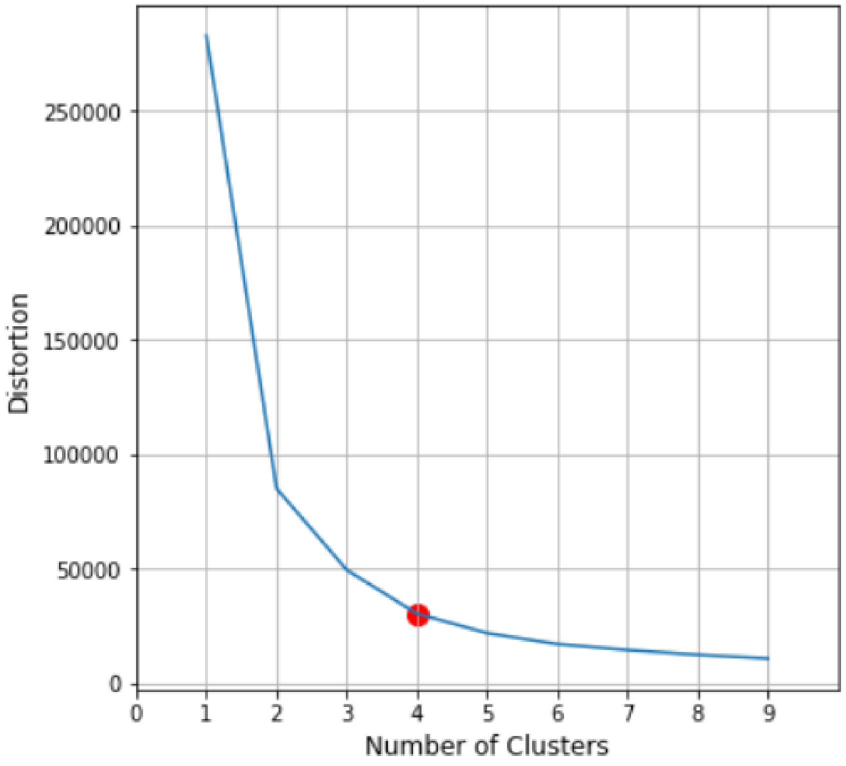


K-means results:

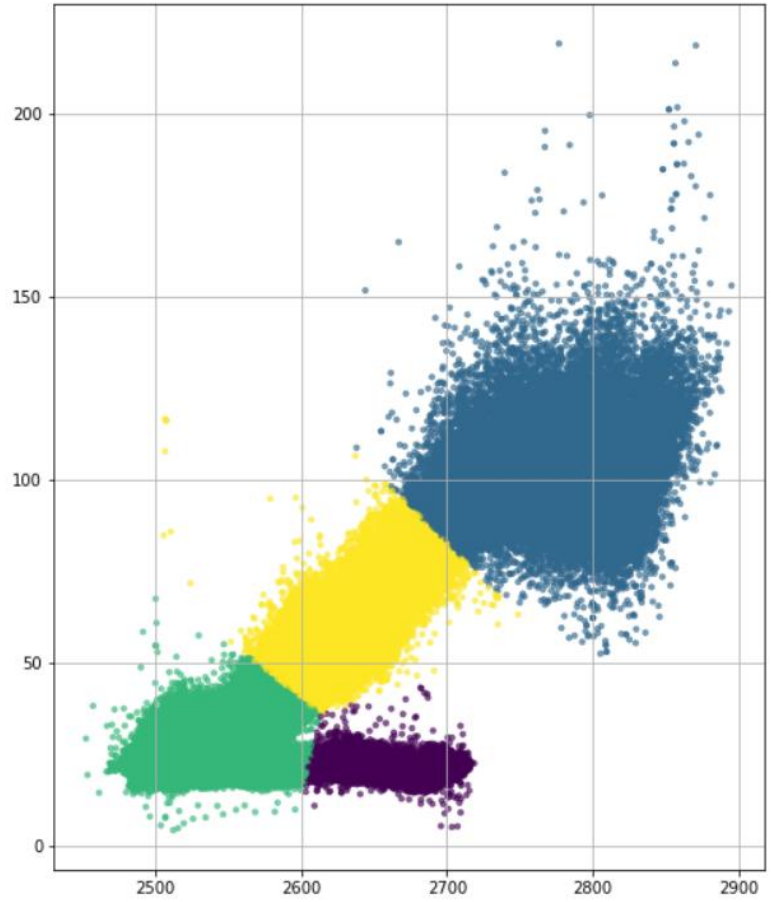
Optimum number of clusters, suggested by the 'elbow' plot:

```
Optimum number of clusters -  4

Out[128]: KMeans(n_clusters=4)
```

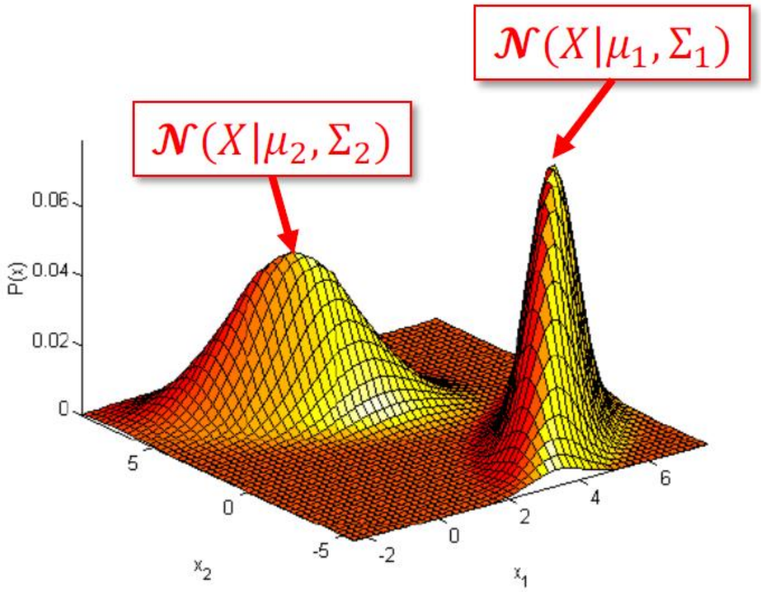


K-means clustering results:



Gaussian Mixture Model (GMM)

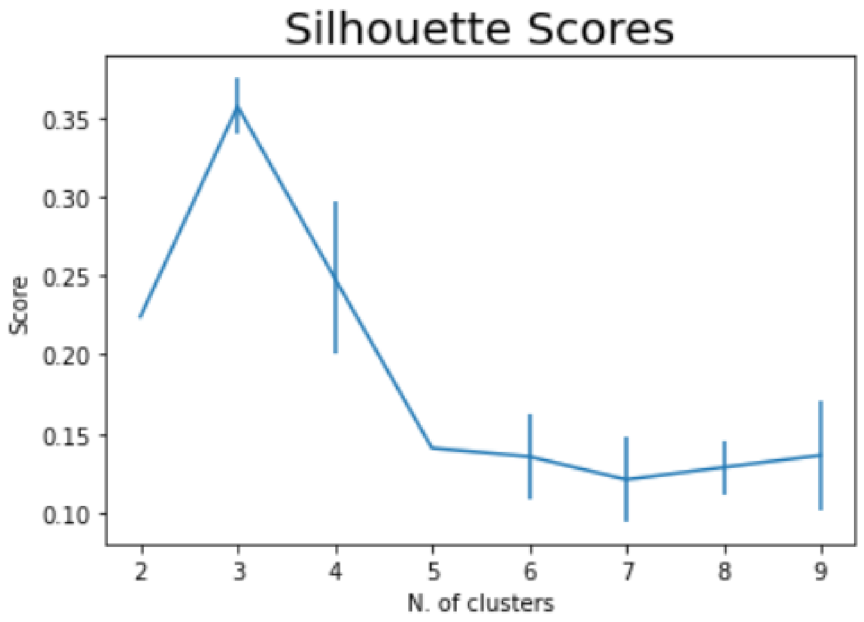
- A density $p(X)$ may be multi-modal, and we can model it as mixture of uni-modal distribution (eg: Gaussians)[2]



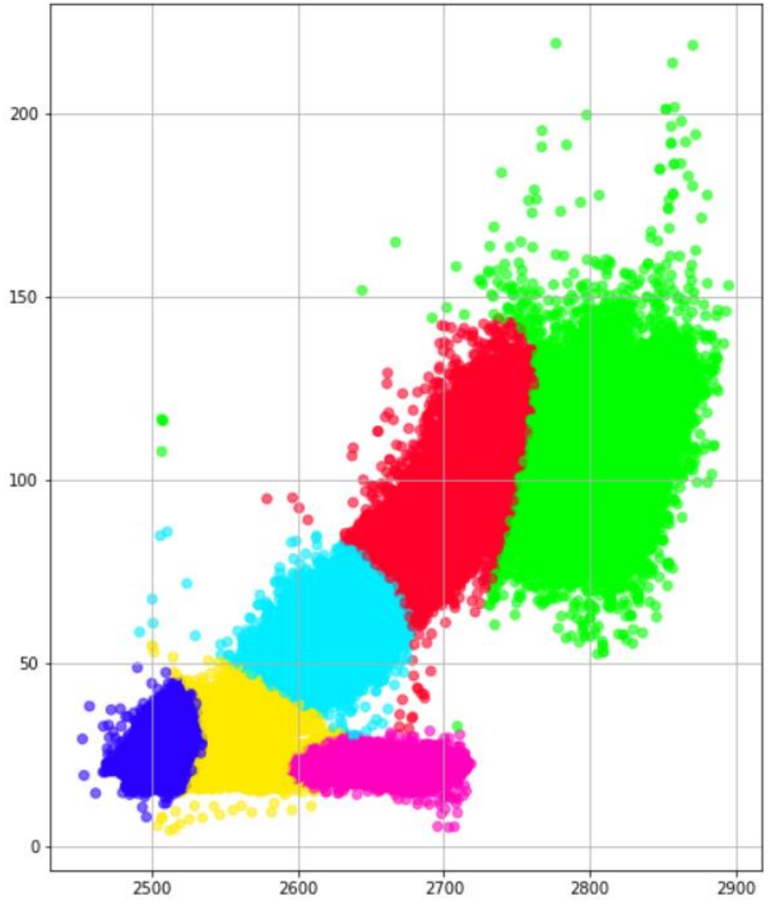
  - $p(X) = \sum_{k=1}^{K} \underbrace{\pi_k}_{\text{mixing proportion}} \underbrace{\mathcal{N}(X|\mu_k, \Sigma_k)}_{\text{mixture Component}}$

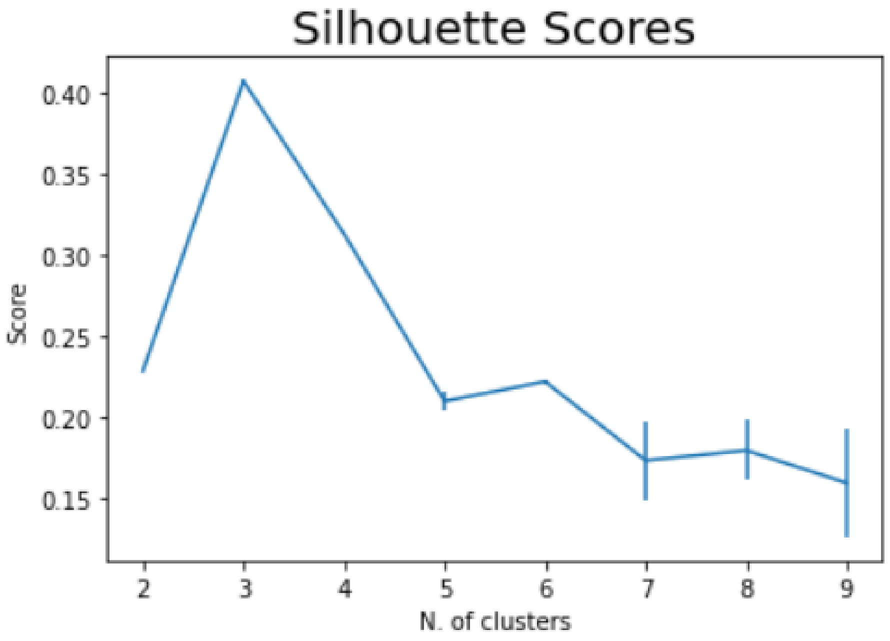
GMM results:

Out[143]: Text(0, 0.5, 'Score')

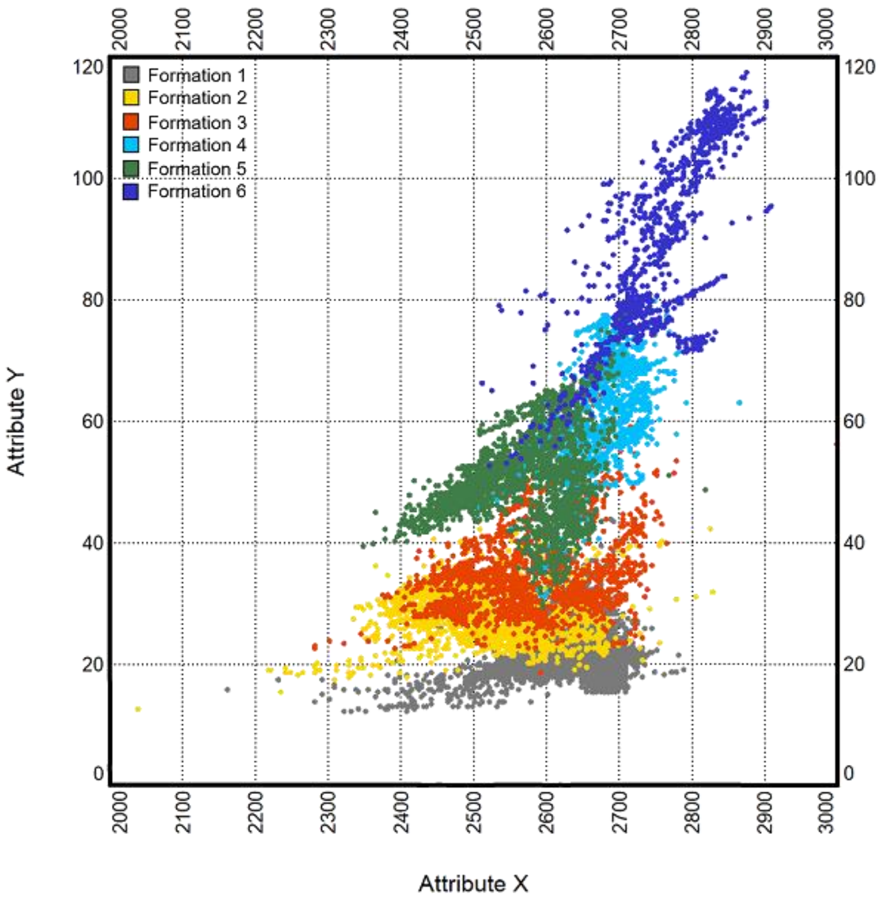


GMM clustering results:





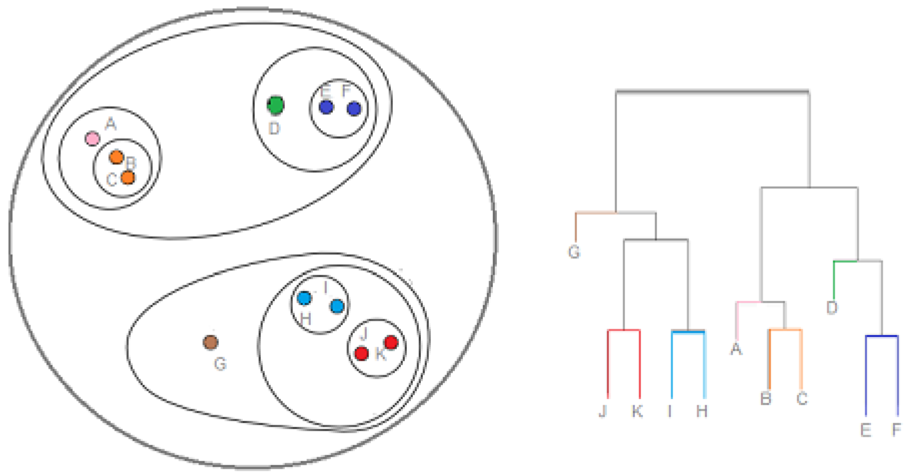
Silhouette Scores
0.40
0.35
0.30
0.25
0.20
0.15
Score
2
3
4
5
6
7
8
9
N. of clusters



Formation 1
Formation 2
Formation 3
Formation 4
Formation 5
Formation 6
Attribute Y
Attribute X
0
20
40
60
80
100
120
2000
2100
2200
2300
2400
2500
2600
2700
2800
2900
3000

Agglomerative Clustering (AC)

- Recursively merges the pair of clusters that minimally increases a given linkage distance[3], each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy [4]

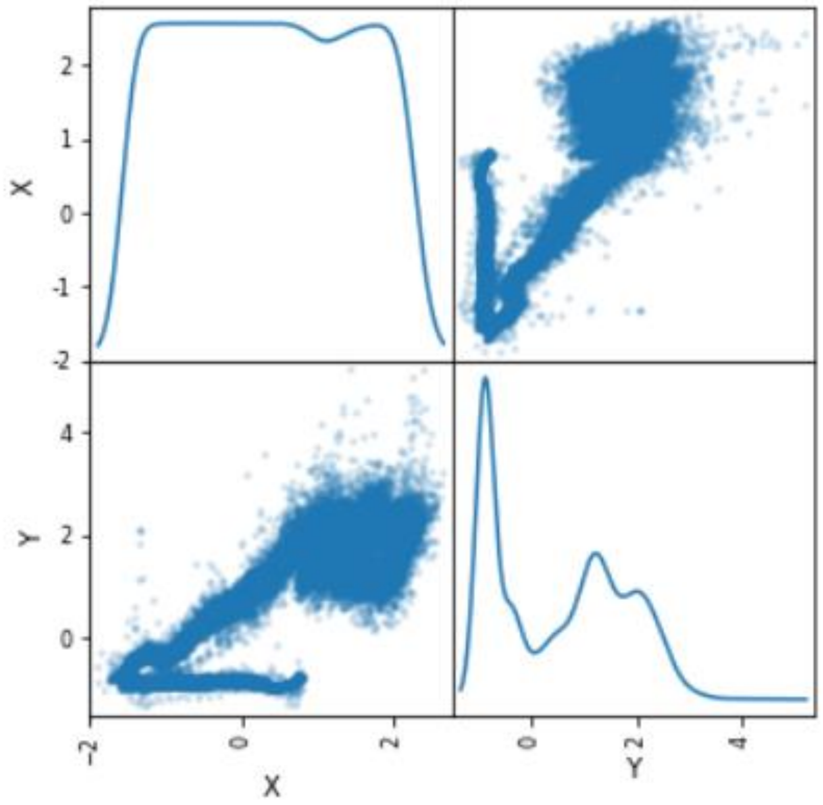


A dendrogram (right) representing nested clusters (left).

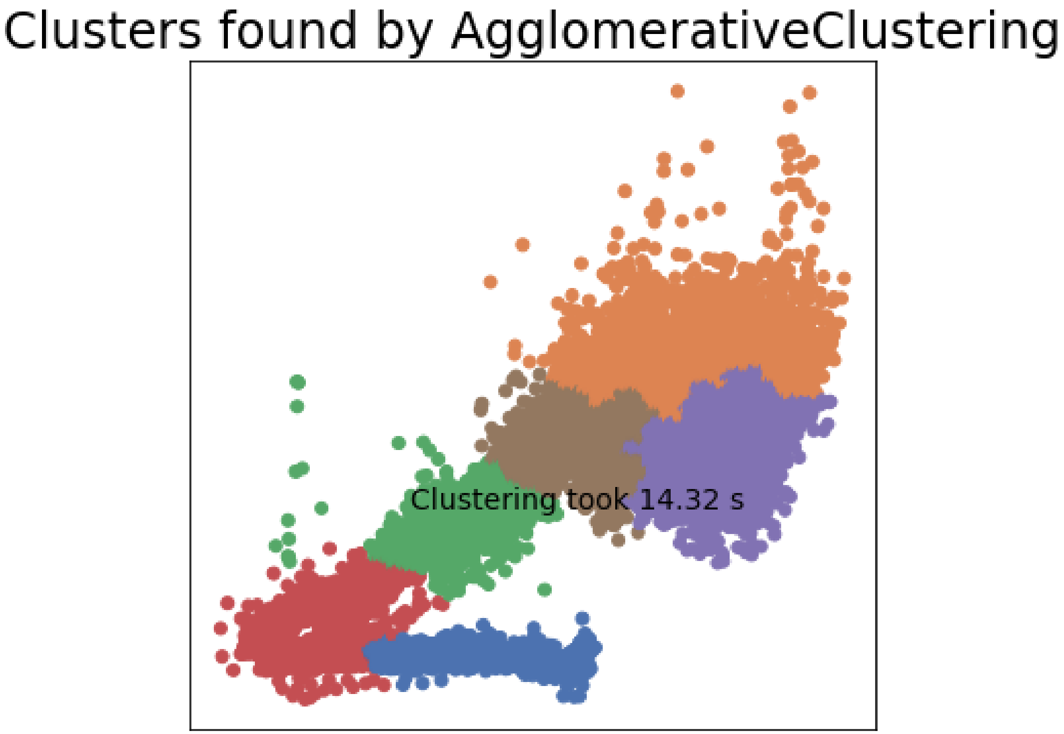
Subsampling before doing AC

- Agglomerative clustering (AC) is a much more CPU intensive technique, than the other two clustering ones, and implementing that on our original dataset, with over 140000 data points, was very time consuming. I have subsampled the original data to be able to demonstrate this technique. While doing subsampling, I attempted to preserve the original signature and distribution of the dataset

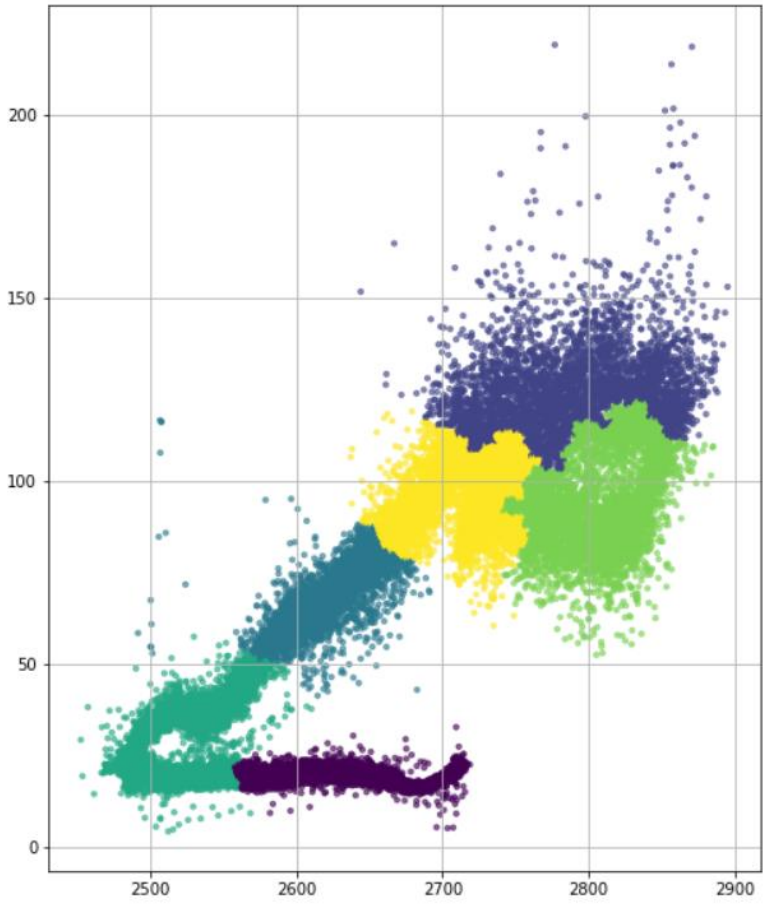
Scatter Matrix After Subsampling



AC results:



AC clustering results:

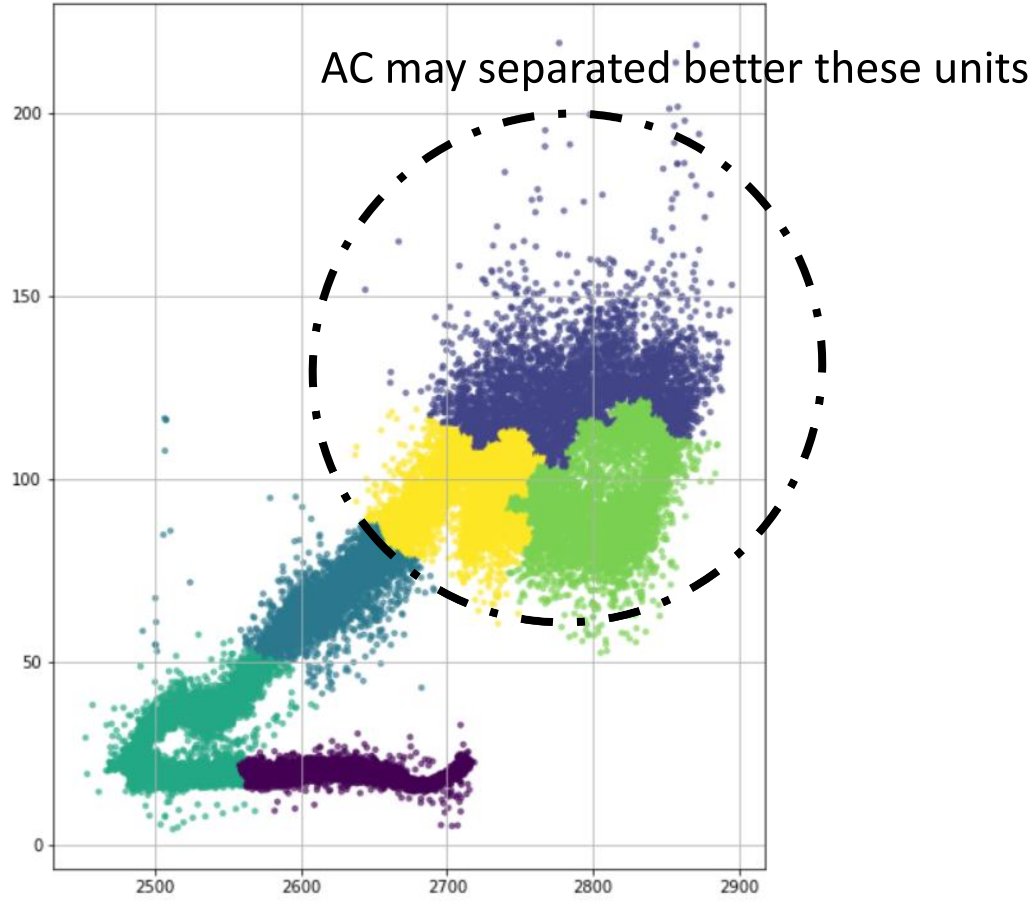


Observation and Conclusion (1/2):

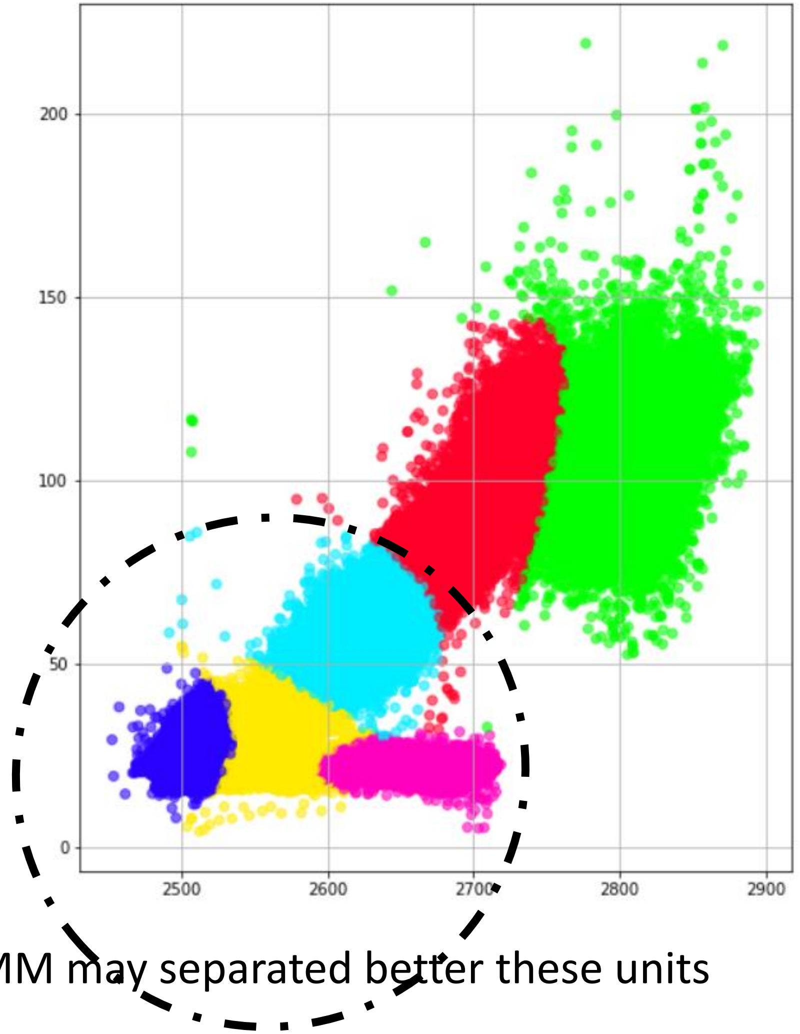
- Three unsupervised clustering methods were implemented on the provided dataset, k-means, GMM and AC.
- K-means works well with the spherical clusters, something that may not be well relevant to the clusters associated with geophysical cross-plots
- I believe GMM is more adoptive approach for geoscience purposes than the K-means
- AC is much more time/CPU consuming approach that the other two methods, and I had to perform data subsampling to perform this method. I however think that this approach may have some potential values in the geoscience routine

Observation and Conclusion (2/2):

AC clustering results:



GMM clustering results:



References

- [1] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
- [2] Georgia Tech Machine Learning course, ISYE 6740, Yao Xie, Ph.D.
- [3] https://en.wikipedia.org/wiki/Hierarchical_clustering
- [4] https://www.statisticshowto.com/hierarchical-clustering/