Analysis of Clustering K-Means and DBSCAN

Akshay Sehgal

***Abstract*—Pattern Recognition helps in solving problems by finding the patterns among incoherent data and make it more coherent. The problem I am analyzing in this report is that of clustering. Clustering gives a very specific and meaningful information about the dataset and lets us draw meaningful conclusions. The prime focus is on K-Means clustering but I have also analyzed DBSCAN clustering to draw comparison between the two and analyzed the failed performance of K-Means. I have also analyzed K-Means for different value of 'k' and analyze the Elbow technique to find the optimal value of k for best results.**

I. Introduction

CLUSTERING is a technique to group data based on different parameters. These parameters are usually the data columns and defines the data. The identical data is grouped in common clusters and the elements/data in each cluster is more related to each other than the elements/data in other clusters. There are various techniques to cluster data and one of the prominent one is K-Means clustering. In this report, I am going to do in depth analysis of K-Means clustering and also do --------- clustering to compare the corresponding results. For K-Mean clustering, it would be easy to visualize dataset for k=2, as it can be plotted in a 2D graph. As we increase the value of k, i.e. add more dimensions/parameters to compare, the visualization becomes difficult.

For example, below are the image how the dataset looks in 2D and 3D mapping. These images are not related to our dataset.

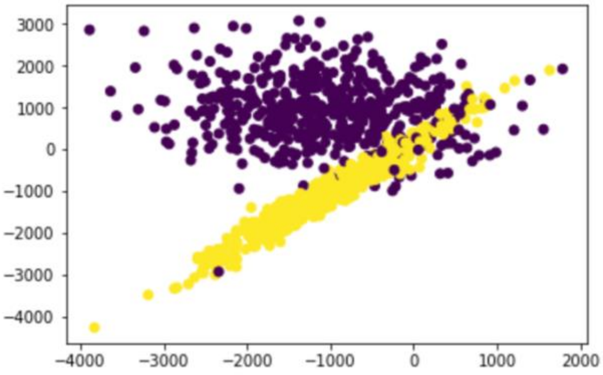


Figure 1 2D Visualization in Clustering

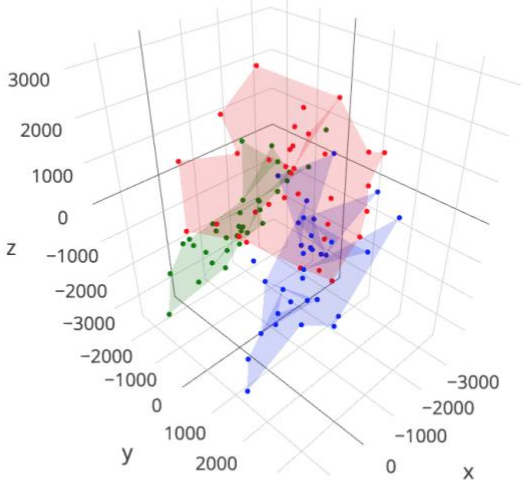


Figure 2 3D Visualization in Clustering

As we can see we can add more dimensions or feature set to improve clustering and have more strictly bounded data but the visualization of the same requires advanced tools.

II. Algorithms

A. K-Means Algorithm

K-Means is widely used clustering technique and is the major study of this paper. It is a technique which employs vector quantization and works iteratively to minimize within cluster variance i.e. Squared Euclidian distances within the cluster. This results in partitioning the datasets into cells which are aligned close to the centroid/mean value. The process is done iteratively to find the optimal value of centroid.

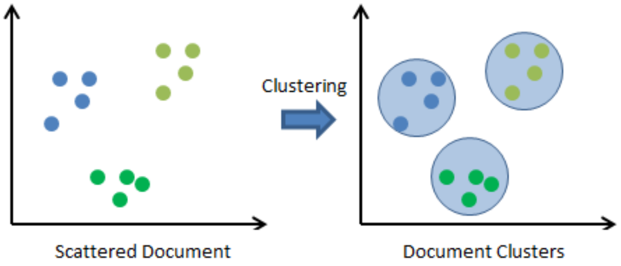


Figure 3 Basic Image describing Clustering

.

Steps involved in K-Mean clustering:
Step 1: Initialize cluster centers
In this step, the algorithm picks up random centroids based on the value of k provided.
Step 2: Assign Values to the closest centroids
For each data point in the dataset, the algorithm calculates distance to all the centroids and assign the data point to the cluster with the centroid with minimum distance.
Step 3: Revise cluster centers
After assigning all the points to the respective clusters, the algorithm updates the cluster centers based on the center of the mass, which can vary from the initialized phase. E.g. in the figure below C1' is the revised cluster center and depends on the spread on the blue cluster. It is same for all other clusters.

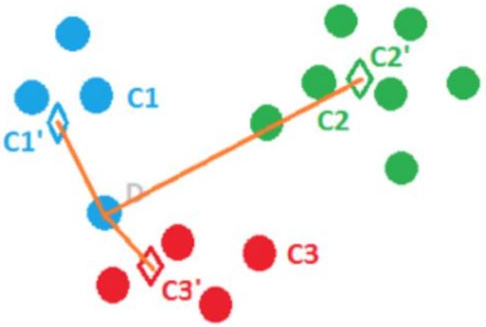


Figure 4 Reassigning the centroids

Step 4: Repeat step 2 and step 3 until convergence
Step 2 and step 3 are repeated until convergence when the Euclidean distance is minimized and finally, we get the solution set.

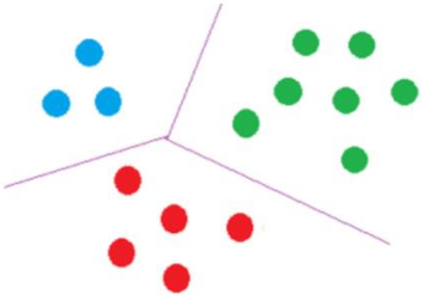


Figure 5 Optimized post clustering result

III. MATH

For K-Means clustering,
If p = (p1, p2) and q = (q1, q2) then the distance is given by

$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

If each cluster centroid is denoted by Ci, then each data point x is assigned to cluster based on

$$\arg\min_{C_i \in C} dist(C_i, x)^2$$

Here dist() is the Euclidian distance.
Finding the new centroid from the clustered group of points

$$C_i = \frac{1}{|S_i|} \sum_{x_i \in S_i}^{n} x_i$$

Si is the set of all points assigned to the ith cluster.

IV. Experiment Stage

Before moving on to the experiment, I would like to provide the information on the datasets that are used to study the clustering algorithms.

BLOB Dataset

BLOB is a utility of sk-learn library of python and is used to generate synthetic data based on the input parameters. I thought of describing blob dataset in order to have better visualization and understanding of the simple dataset before moving on to higher order or larger datasets. Using the blob dataset, we already know about the dataset and the corresponding clusters. K-Means clustering would help us verify the truth and help us understand the basics of clustering.
For this I chose to generate dataset of 300 samples with 2 centers and ran K-Means to find the position of centroids.
Figure shows the blob dataset plot.

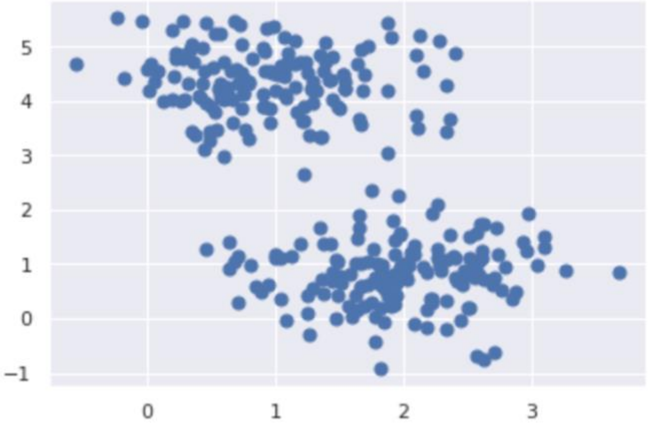


Figure 6 BLOB Dataset

WINE Dataset

The wine dataset is selected from UCI repository and the link is here. The dataset comprised of 4898 datapoints with 11 features.

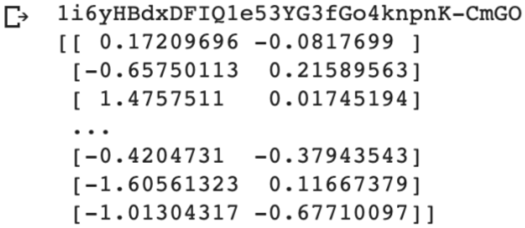
BIRCH Dataset

In the final attempt to plot a large dataset using K-Means. I selected Birch Dataset from here. This is a large dataset of around 100K records and can be visualized using K-Means. I selected random size cluster data to see the non-symmetrical patterns that are generated in the dataset using clustering.

1. Initialization

The initialization phase of the clustering involves reading the data and transform it to a new array with the same number of samples and new feature set. This data is then used for the clustering. I used StandardScaler.fit_transform() method of which is part of sklearn.preprocessing library. This method scales down the data and make it ready for processing.
The image below shows the transformed data for white wine

dataset. The initialization is random for K-Means clustering and can take a lot of number of iterations to converge, whereas this has been improved in K-Means++ where centroids are assigned based on the spread of dataset and converges soon.



```
1i6yHBdxDFIQ1e53YG3fGo4knpnK-CmGO
[[ 0.17209696 -0.0817699 ]
 [-0.65750113  0.21589563]
 [ 1.4757511   0.01745194]
 ...
 [-0.4204731  -0.37943543]
 [-1.60561323  0.11667379]
 [-1.01304317 -0.67710097]]
```

2. Elbow Method

The Elbow method enables us to select the optimum value for the number of clusters. This it does by fitting the model with the range of value of K and calculating the Sum of Squared Euclidean distance and plot it on the graph for each value of k from 1 to the upper limit. By plotting the data for this range, we can find the point of inflection/Elbow in the dataset and this is usually the optimal value of k. As shown in the figures below. I have chosen the range to be this. Figure shows the inflection at k=2, this is for the blob dataset. Figure shows the inflection at k=3, this is for the wine data. So, I guess now we have a fair understanding of how the Elbow technique enable us to select the optimal value for K-Means clustering from the given range.

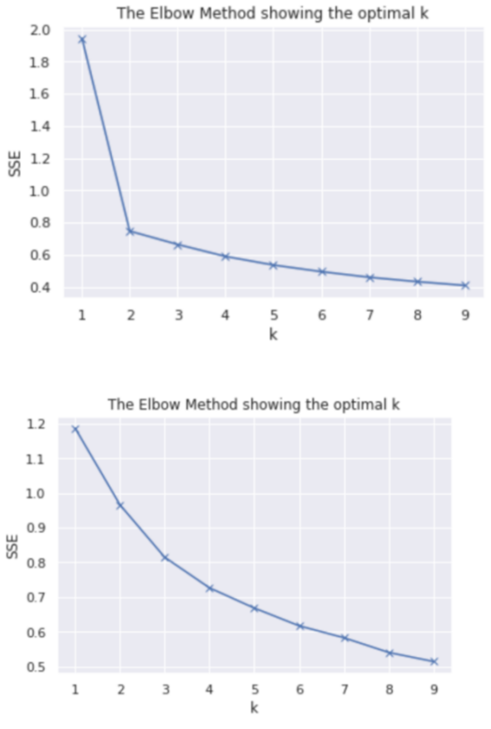


Figure 7 Elbow Method Results

3. Post Clustering Results

Post clustering results show the position of cluster with the corresponding centroids.

The figure below shows the clustering on blob dataset.

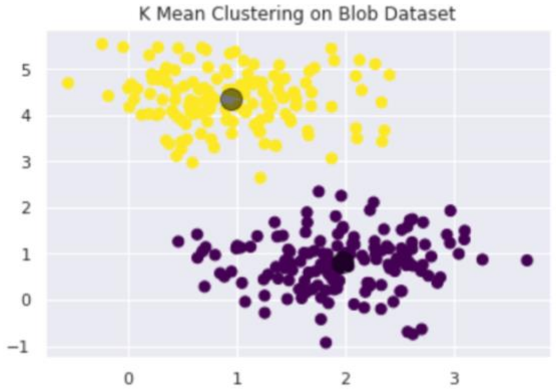


Figure 8 Post Clustering results for BLOB Dataset

Experiment 1

Clustering of Large Dataset (Birch Dataset)

K-Means clustering is done on the Birch dataset and the results obtained best clustering for k=100. It is also analyzed for k=10.

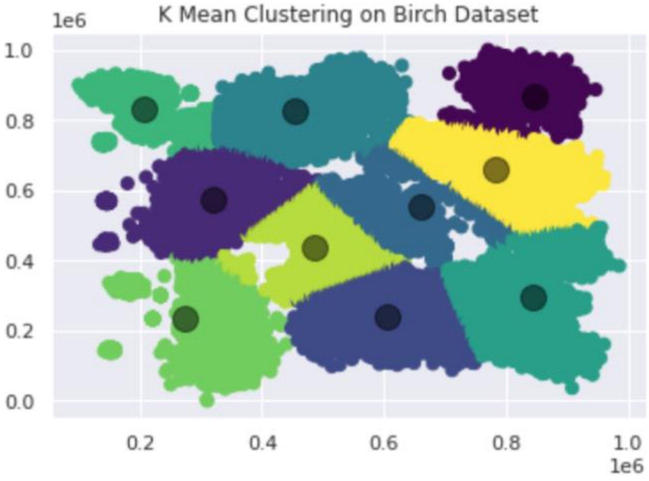


Figure 9 Large Dataset clustered for K=10

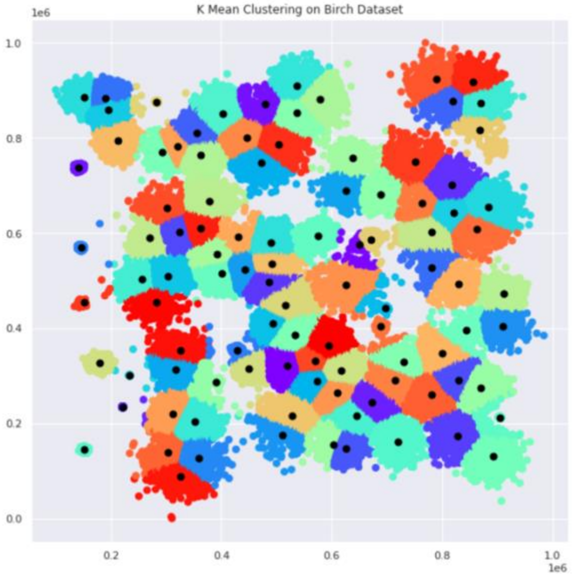


Figure 10 Large Dataset clustered for K=100

Experiment 2

This experiment enables us to find the accuracy of the K-Mean clustering, as the data is already labeled. But we can use this data without labels and this clustering problem can be converted to classification problem and can be checked for its accuracy. Since K-Means follow heuristics approach, the accuracy value may vary, so I have selected the average of 5 results to depict the average accuracy.

The wine dataset has 11 features and out of which Quality of wine is the feature taken for consideration to cluster similar quality wines together and differentiate between bad, not bad and good quality wines.

The figure below shows the classification with respect to quality for all the dataset.

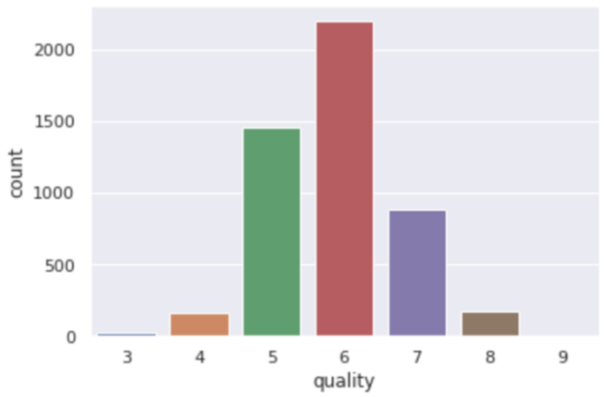


Figure 11 Plot for Quality vs Count

The wines are distributed into 3 qualities, depicted here as 0(bad), 1(normal) and 2(good). With respect to data 2 is assigned to good quality which is greater than 8. 0 for bad quality which is less than 5 and the rest are assigned value 1.

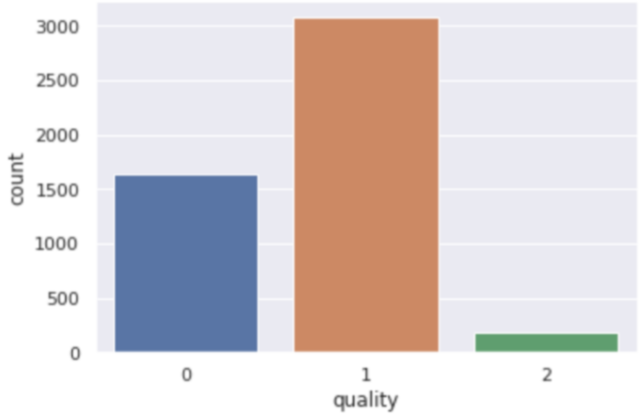


Figure 12 Plot for New Labels for quality

Then we used K-Means implementation to confirm the accuracy for the transformed data with three labels changing it into a three-class cluster problem. The average accuracy came around 71.69%.

The result of K-Means for white wine data is shown below. The placement is not clear because the data contains 11 features and is not placed in a 2D frame.

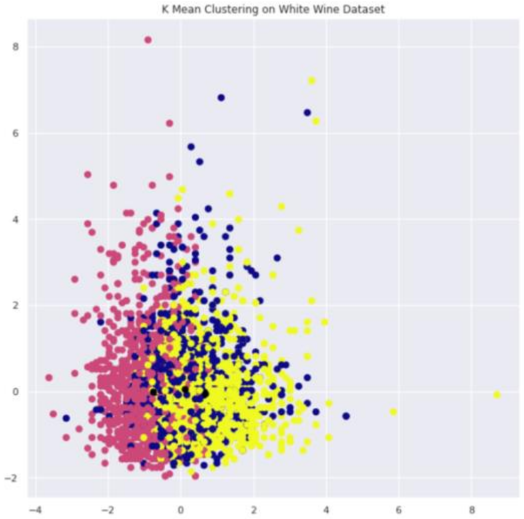


Figure 13 Clustered result for Wine Data for K=3

Experiment 3

In experiment three, K-Mean clustering is done on moon shaped data. The aim of this experiment is to see the kind of shape allocation done by the K-Mean clustering and to figure out any efficient clustering algorithm that can help identify shape in a more efficient manner. The alternative algorithm is explained later. The figure below shows the moon shaped data.

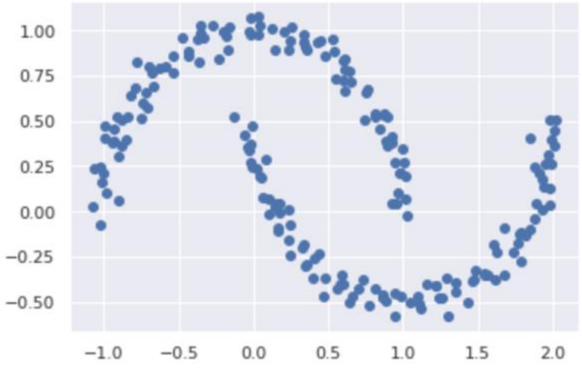


Figure 14 Moon Shaped Data Set

After applying K-Means algorithm on the above dataset for value of k=2 (Elbow Method), the following cluster is obtained.

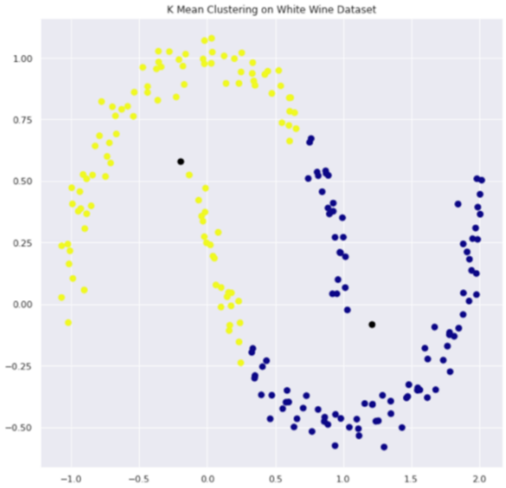


Figure 15 Clustered Using K-Means

As can be seen from the above clustering pattern, K-Means cluster data in a spherical manner considering Euclidean Distance from the initialized centroids which are then optimized to have the minimum sum of Euclidean distances. For shapes like moon shape example, the algorithm is unable to consider the connected components and as a result close data points in the neighbor can be considered in a separate cluster.

This leads us to exploring another clustering algorithm which is known as DBSCAN (Density based spatial clustering of Application and Noise). DBSCAN works based on the density of the data points, the parameters provided are esp – the maximum distance for two points to be considered as neighbors and hence belong to the same cluster and min_samples – the number of samples in the neighborhood for that point to be considered as a core point.

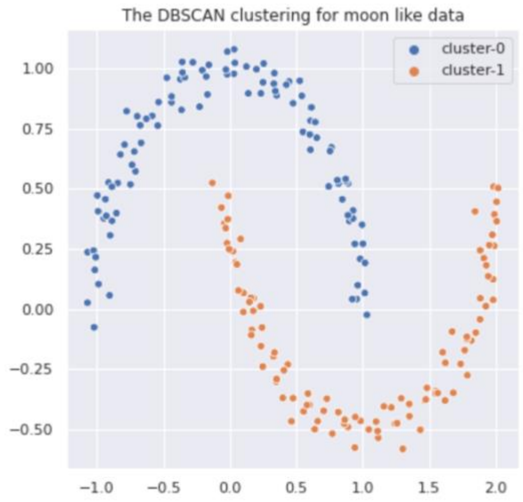


Figure 16 Clustering using DBSCAN

DBSCAN works well with the noise in the dataset. The example is shown below.

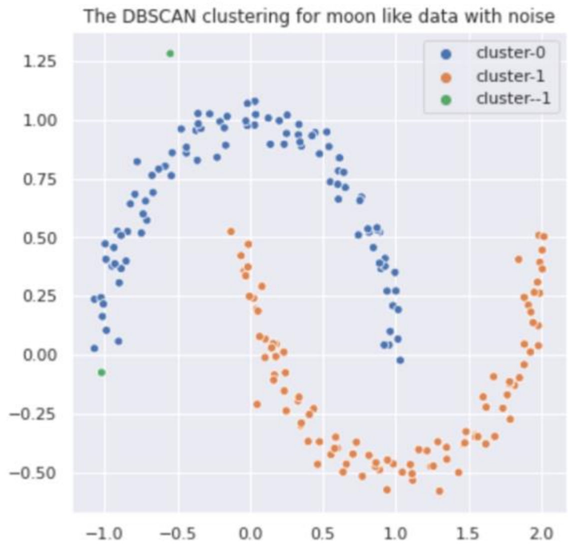


Figure 17 Clustering using DBSCAN with Noise

As you can see it identifies the noise as green data points as cluster – 1. To make noise points more prominent the value of ‘esp’ is reduced from 0.3 to 0.2.

V. Results

1. For experiment 1, it is observed that for small values of K i.e. 10, the cluster size is huge and may include a lot of points which should not be classified under the same cluster. This points to the fact to find the right value of K and it can be a tedious task to find the value of K beforehand for large datasets and real time data.
2. From Experiment 2, it is observed that K-Means clustering can work for N-dimensional data sets, though plotting and visualizing the cluster in 2D plan is difficult. K-Means can be used to classify data with the accuracy of approx. 70% and can vary depending on centroid initialization. It may not be sufficient for some application.
3. From Experiment 3, it is observed that the K-Means clustering because of its nature to find clusters on the basis of Euclidian distance always work in spherical zone. It does not consider the data on the probabilistic basis, density or of varying sizes.

   In continuation to this, DBSCAN clustering algorithm is studied which is able to detect data of varying sizes and also able to filter out noises efficiently. This it does based on density and proximity of data points.

VI. Conclusion

In conclusion, with all the experiment and observation K-Means has its advantages and disadvantages.

Advantages of K-Means are:

- Relatively issue to implement
- Scales to large data sets
- It does guarantees convergence
- Generalize the clusters of different shapes

Disadvantages of K-Means are:

- Finding the optimal value of k manually
- Heavily relying on initial values of centroids
- Clustering for datasets of varying density and sizes is difficult
- It does not scale with number of dimensions
- Centroids can be dragged by outliers that is data points that are really far can pull the cluster of get its own cluster instead of being ignored.

References

[1] https://www.bigendiandata.com/2017-04-18-Jupyter_Customer360/

[2] https://en.wikipedia.org/wiki/K-means_clustering

[3] https://healthcare.ai/step-step-k-means-clustering/

[4] https://muthu.co/mathematics-behind-k-mean-clustering-algorithm/

[5] https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages

[6] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans.predict

[7] https://jakevdp.github.io/PythonDataScienceHandbook/05.12-gaussian-mixtures.html

[8] https://towardsdatascience.com/

[9] https://www.scikit-yb.org/en/latest/api/cluster/elbow.html

[10] http://cs.joensuu.fi/sipu/datasets/

[11] https://pythonprogramminglanguage.com/kmeans-elbow-method/

[12] https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html
[13] https://www.bigendiandata.com/2017-04-18-Jupyter_Customer360/
[14] https://medium.com/code-to-express/k-means-clustering-for-beginners-using-python-from-scratch-f20e79c8ad00
[15] https://www.kaggle.com/vishalyo990/prediction-of-quality-of-wine