Unit 6: Clustering with Python

This unit focused on clustering with python more specifically K-Means Clustering. Wu (2021) describes K-Means Clustering as an unsupervised learning algorithm used to partition a dataset into K distinct, non-overlapping subsets or clusters. The algorithm’s aim is to minimise the variance within each cluster, making the points in each cluster as similar as possible.

K-Means is an iterative algorithm that assigns each point to a cluster with the closest centroid. The centroid of these clusters is then calculated again by taking the average. The steps involved are as discussed below:

1. **Initialisation**: This involves selection of K initial cluster centroids randomly from the dataset. These centroids can be chosen randomly or by using some heuristic.
2. **Assignment Step**: Each data point is assigned to the nearest centroid. This creates K clusters based on the distance between data points and centroids. The distance is typically measured using **Euclidean distance** (Raschka et al., 2022).
3. **Update Step**: The new centroids are calculated as the mean of all data points assigned to each cluster. The centroid of a cluster is updated to the average position of all points within that cluster.
4. **Repeat Steps 2 and 3**: All data points are reasssigned the nearest centroid based on the updated centroids and then centroids recalculate. This iterative process continues until the centroids no longer change significantly, indicating that the algorithm has converged.
5. **Termination**: The algorithm stops when the centroids have stabilised, that is, their positions do not change significantly between iterations or after a predetermined number of iterations.

**Advantages of K-Means:**

* Simplicity and Ease of Implementation: K-Means excels in its simplicity and ease of implementation. The algorithm is straightforward and easy to implement.
* **Computational Efficiency:** K-Means is highly efficient in terms of computational complexity, especially for large datasets. The algorithm's time complexity is O(n \* k \* t), where n is the number of data points, k is the number of clusters, and t is the number of iterations, making it scalable and suitable for big data applications (Gupta et al., 2021).
* **Effectiveness with Well-Separated Clusters:** K-Means performs exceptionally well when clusters are well-separated and spherical in shape. It excels in scenarios where the dataset has distinct, non-overlapping clusters, providing clear and interpretable results.
* **Versatility and Flexibility:** K-Means can be applied to various types of data and can handle different scales and dimensions. It is versatile enough to be used in numerous fields such as image segmentation, market segmentation, document clustering, and more.
* **Ease of Understanding and Interpretation:** The results of K-Means clustering are easy to understand and interpret. The centroids represent the center of each cluster, and the data points within each cluster are similar to each other, making it intuitive to analyze and visualize.

**Limitations of K-Means:**

* **Pre-specification of K:** The number of clusters (K) must be specified in advance, which may not always be known. Determining the optimal number of clusters often requires additional methods such as the elbow method or silhouette analysis, adding complexity to the process.
* **Sensitivity to Initial Centroid Placement:** The initial placement of centroids can significantly affect the final clustering outcome. Poor initialization can lead to suboptimal solutions. Methods like K-Means++ can be used to improve the initialization process, but the issue remains a challenge (Bahmani et al., 2012).
* **Local Minimum Convergence:** The algorithm may converge to a local minimum rather than the global minimum, depending on the initial centroid positions. This can result in less optimal clustering solutions. Multiple runs with different initializations are often necessary to find a more optimal solution.
* **Assumption of Spherical and Equally Sized Clusters:** K-Means assumes that clusters are spherical and of equal size, which may not hold true for all datasets. This assumption limits its effectiveness for datasets with clusters of irregular shapes or varying sizes.
* **Difficulty Handling Noise and Outliers:** K-Means is sensitive to noise and outliers, which can distort the cluster centroids and adversely affect the clustering performance. Outliers can disproportionately influence the centroid calculation, leading to incorrect clustering results (Gupta et al., 2021).
* **Scalability Issues with Very Large Datasets:** While K-Means is generally efficient, handling extremely large datasets with high dimensionality can still pose scalability challenges. Advanced techniques and optimizations, such as mini-batch K-Means, may be necessary to manage very large datasets effectively.

See Jupyter Notebook for this unit’s activity here:

**References**

Bahmani, B., Moseley, B., Vattani, A., Kumar, R. & Vassilvitskii, S., (2012) Scalable K-Means++. DOI: https://doi.org/10.48550/arXiv.1203.6402

Gupta, A., Sharma, H. & Akhtar, A. (2021). A comparative analysis of k-means and hierarchical clustering. *EPRA International Journal of Multidisciplinary Research (IJMR)* 412–418. DOI: http://dx.doi.org/10.36713/epra8308

Raschka, S., Liu, Y. & Mirjalili, V. (2022). 'Working with Unlabeled Data – Clustering Analysis' In: Raschka, S., Liu, Y. & Mirjalili, V. Machine Learning with PyTorch and Scikit-Learn. Packt Publishing.

Wu, B. (2021). 'K-means clustering algorithm and Python implementation' In: 2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE). IEEE. 55–59.