K-Means Clustering: How Initialisation and the Choice of K Shape the Results

Github Repositry: https://github.com/satwik1704/k_means

Abstract

In unsupervised machine learning, clustering is a primary method for finding the structure and patterns within unlabelled data.K-Means clustering methods are widely used due to being simple to implement, having fast execution time, and producing easily interpretable outputs/results. However, K-Means has two major downsides: the way initial centroids are selected and the way K is chosen. If the initial centroid selections or K values are not appropriate, this can lead to poor quality/solutions produced by K-Means.

This report will provide a complete tutorial of how K-Means works and then examine the initialisation and K values that impact the way K-Means operates. We will use Python to demonstrate the centroids' movement and demonstrate the difference between random initialisation and K-Means++, and to show the results of applying both the Elbow method and Silhouette score to determine the appropriate value for K. Additionally, we will cover a case study of K-Means performing poorly with a non-spherical dataset and briefly explain when to use K-Means and when not to use it. The report will combine theoretical explanations of clustering algorithms with practical demonstrations of how clustering is achieved and searching for clustering solutions will demonstrate many of the generic methods and models used in machine learning, including distance metrics, iterative optimisation, and clustering by centroids.

1. Introduction

Data without labels is common in many everyday scenarios, for example, you may have thousands of customer records and images, or sensor data, but have not created specific categories for each record/image. Clustering groups similar observations so that the user can look for different structures/patterns as well as conduct exploratory analysis for downstream model building.

K-Means clustering is one of the first algorithms that are often used for clustering. K-Means partitions data into K clusters, where K is the user-supplied 'number of clusters,' by assigning each point of data (observation) to the cluster with the nearest point/centroid. The algorithm is simple to conceptualise and is able to scale well to larger datasets than other clustering algorithms, thus allowing for ease of use when visualising the results when the data is in a low-dimensional space.

The Number of Clusters is Not Specified. To execute the K-Means algorithm, the user must designate a number of clusters, or K. The final grouping of data (known as the final cluster) may depend on the position of the initial centroid(s).

This paper will:

- To offer a systematic overview of the K-means clustering algorithm.
- To demonstrate how the positions of the centroids affect cluster behaviour.
- To provide suggestions for determining the optimal number of clusters K using both the elbow method and the Silhouette method.
- Present a failure case where K-Means will not work for the structure of the data.
- When possible, link these concepts to existing methodologies in generic machine learning such as distance-based learning and iterative optimising.

2. Clustering and Unsupervised Learning

Clustering belongs to the category of unsupervised learning in the sense that clustering does not have predetermined labels associated with the data. Rather than using labels to determine the associated class of a new input sample, algorithms using clustering determine how to best represent or group the input samples based upon their similarity to one another.

Some of the more widely used clustering algorithms are:

- Centroid-based clustering (K-Means) where a singular centre represents a cluster.
- Density-based clustering (DBSCAN) where clusters are defined as regions of high density of points in the data space.
- Hierarchical clustering, which is built around clustered trees, similar to a family tree representation.
- Probabilistic clustering (Gaussian Mixture Model) clusters are treated as probability distributions.

K-Means is a centroid based clustering algorithm where Euclidean distance is the metric used to determine the similarity between points. K-Means is greatest when compact, round clusters are formed using similarity metrics and multiple generic clustering algorithms provide additional data refinement through distance measures, iterative data refinements and minimising variances.

3. The K-Means Algorithm

The dataset X = {x_1, ..., x_n} consists of points in the d-dimensional real space ($\mathbb{R}^d$), and we want to cluster them into K clusters C_1, C_2,..., C_k. These clusters each have a centroid μ_j, and the goal of K-means is to minimize the sum of the squares of the distances between all points in a cluster and the cluster's centroid (within-cluster distance).

The objective function for the K-means algorithm is:

$$J = \Sigma_{j=1}^{K} \Sigma_{xi} \in C_j \, ||x_i - \mu_j||^2$$

The algorithm can be summarized with the following steps:

1. Initialization

Select K initial centroids ($\mu_1, \mu_2, ..., \mu_k$).

2. Assignment Step

For each data point, assign it to the cluster whose centroid is closest to the point:

$$cluster(x_i) = argmin_j \{ ||x_i - \mu_j|| \}$$

3. Update Step

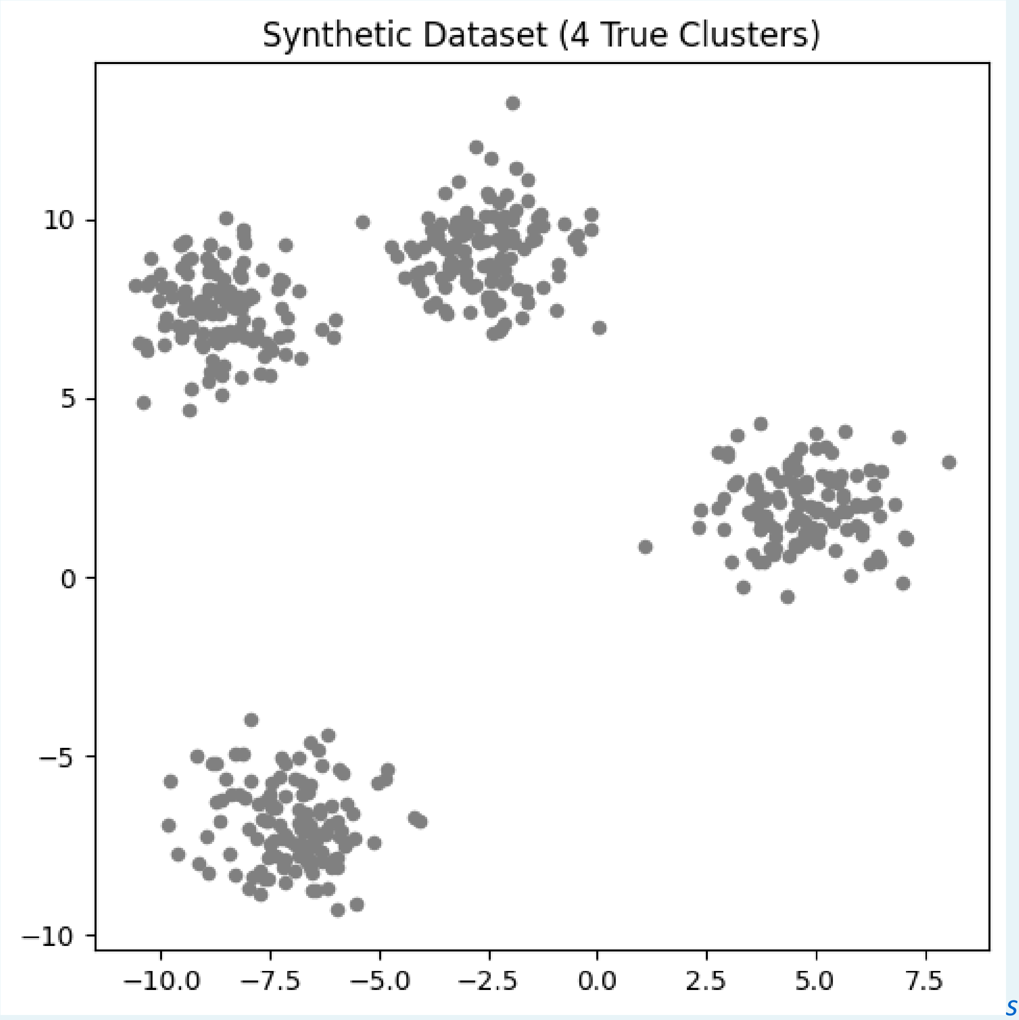
Calculate the new centroid for each cluster by calculating the average of the assigned points:

$$\mu_j = (1/|C_j|) \Sigma_{xi} \in C_j \, x_i$$

Convergence

As long as the cluster centroids continue to change substantially from iteration to iteration, the algorithm will repeat the process of assigning data points to the cluster centroids and then updating the location of the centroids. When there is no longer a significant change in the cluster centroids' locations or after a specified maximum number of iterations has been reached, K-Means terminates.

At each iteration of K-Means, an attempt is made to minimize the K-Means objective function, J; however, K-Means is an optimization technique using a series of iterative steps to minimize J. Since no assurance exists that the achieved result of K-Means will represent a globally optimal value (the least value of J), the K-Means result is not considered a global minimum, but a locally optimal value. Therefore, the proper initialization of the K-Means clustering process is necessary.



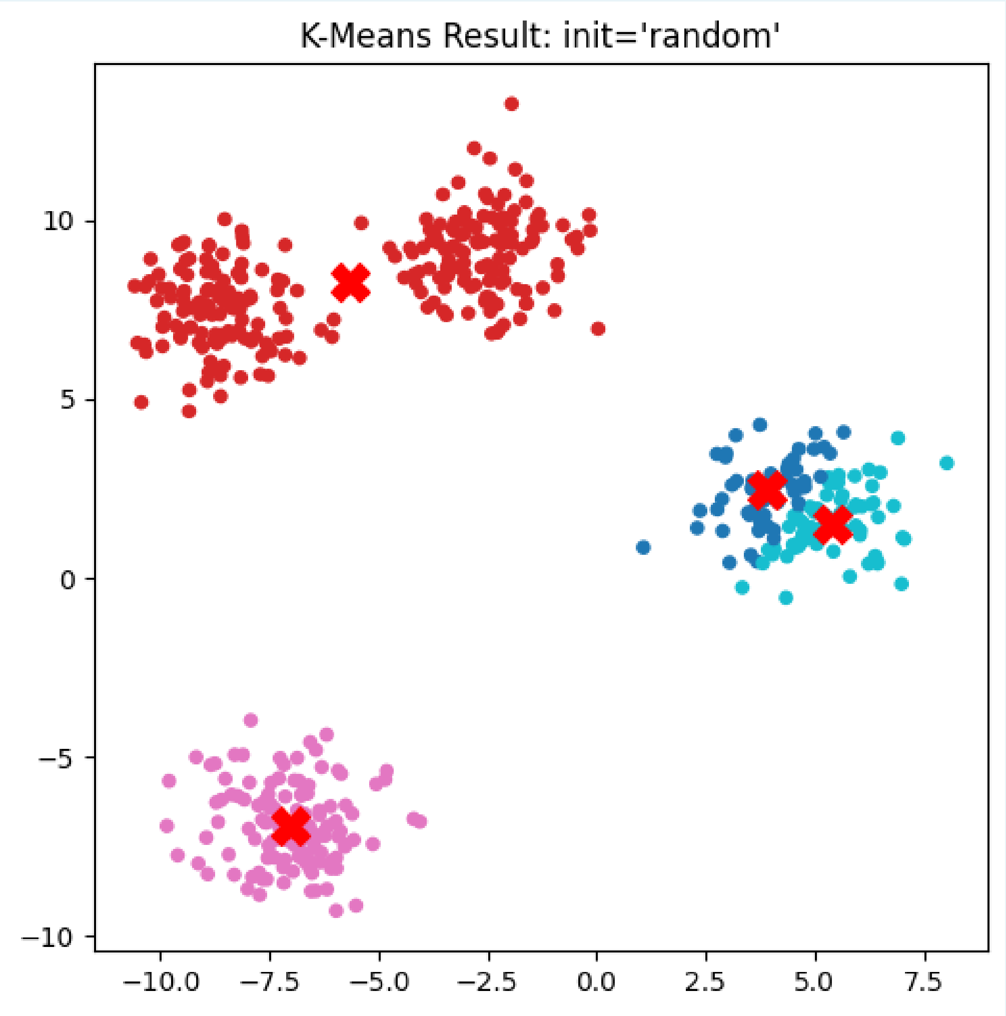
synthetic dataset used for clustering

The example synthetic dataset that we will be using for most of this tutorial consists of 500 data points created around 4 true cluster centroids.

4. Initialization Methods: Random Initialisation or K-Means++

4.1 Random Initialization

Randomly selecting K points as initial centroid positions is the simplest way to initialize K-Means, as this is straightforward to implement. If centroids are chosen incorrectly, the outcome of k-means clustering may be to create uneven clusters. The clustering process is greedy, meaning that as clusters are formed, any underrepresented areas will become difficult to cluster, and may become trapped in local minima.



K-Means clustering with random intialisation

As shown in Figure 2, random initialisation produces clusters that do not align with the underlying data. This is illustrated by the K-Means centroids positioned slightly off centre and the incorrect clustering of certain points in the dataset into the wrong cluster.

4.2 K-Means++ Initialisation

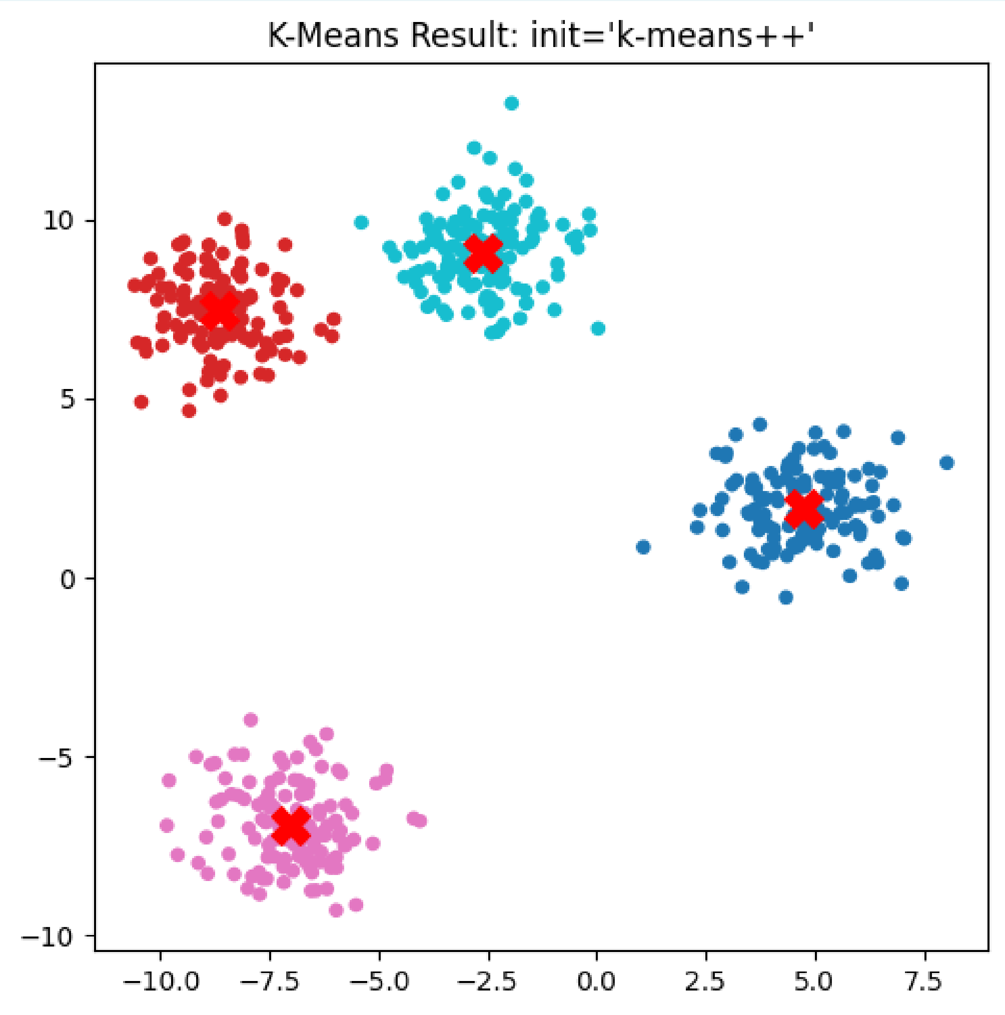
The K-Means method has been improved on with the use of K-Means++, which uses a different way of choosing the first set of Centroids (cluster centres).

When using K-Means++ the following is the method of seeding a new set of Centroids:

1. Randomly choose the first Centroid from the available Data Points.
2. To select the next and subsequent Centroids, the usable Data Points are given a weighted selection likelihood based on the square of the distance to the closest Centroid.

The further away from existing Centroids a Data Point is, the higher the likelihood of the Data Point being selected.

As a result of the Centroids being selected in this fashion, the Centroids are positioned far apart in space initially (well spread out).



K-Means Clustering with K-Means++ Initialisation

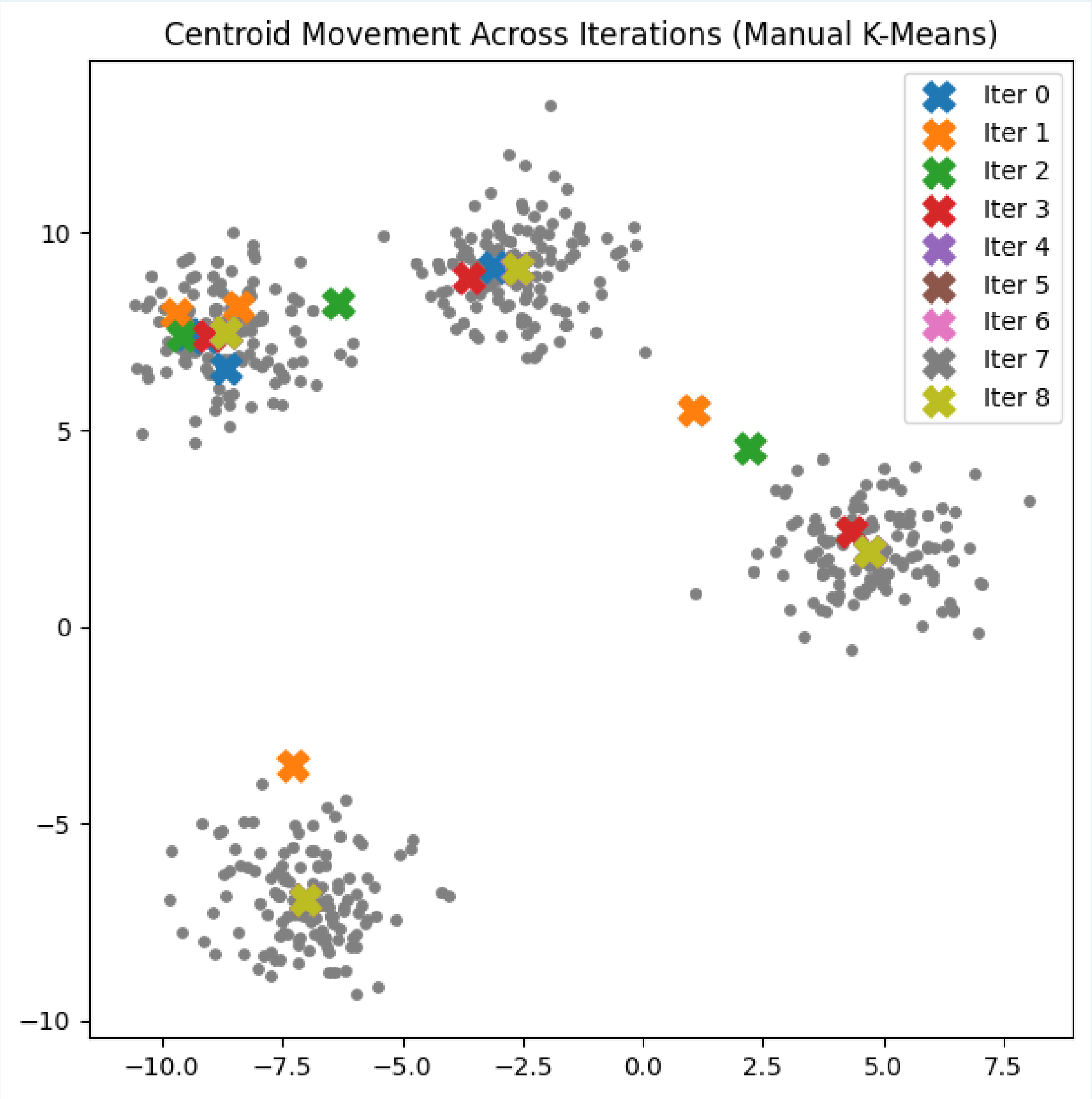
K-Means++ uses better initial Centroid positioning to result in a cleaner and more consistent clustering solution. K-Means++ starts the centroids from decent initial positions, enabling the algorithm to reach a "good" solution quickly. Most libraries, such as those in scikit-learn, use K-Means++ by default as the way to choose the first centroid positions.

From a methodological perspective, Initialisation is a very effective optimisation technique. It is a generalisation technique for producing better quality Local Minima for Algorithms.

5. Visualizing the Movement of the Centroids

The K-Means clustering algorithm is an excellent instructional tool because we are able to see the path that the centroids take from one iteration of the algorithm to the next. If we manually take the time to execute several iterations of the K-Means procedure, we could create a graph that shows us:

1. The points contained within the data set
2. The position of each centroid at each iteration of the K-Means algorithm
3. The movement of centroids over iterations



Centroid Movement Across Iterations

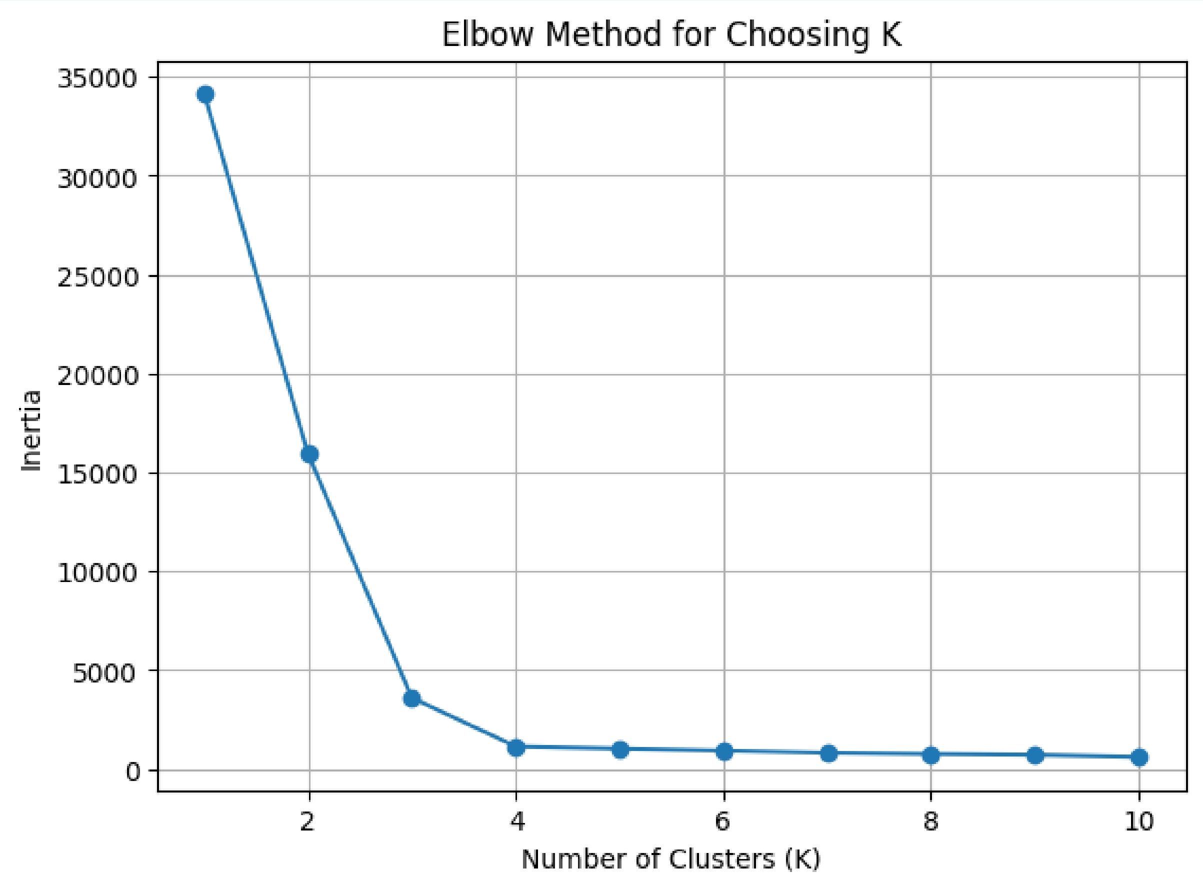
Figure 4 is designed to help illustrate the change in centroid position through time in the context of approximately eight iterations of running K-Means. By examining Figure 4 we can clearly see that the earliest positions of centroids are often much further away from their "true" centre than they are at the conclusion of the iterations. However, as we complete additional iterations of assigning points to clusters and updating the centroid positions, we can see that this movement naturally causes centroids to migrate toward the denser clusters of points. This is especially evident when looking at the locations of centroids in the early portions of the K-Means iterative process. When beginning an algorithm, K-Means begins with many large position movements followed by more precise adjustments as it gets closer to the final position or solution. This is very much how many of the algorithms operate or optimize their ability.

6. The "K" Value Selection in K-Means Clustering

The number of clusters (K) must be specified by the user. The placement of K in this case is a problem of model selection. The two most commonly used methods of selecting K are the elbow method and silhouette analysis.

6.1 Elbow Method

For each possible value of K, we run K-Means and collect J, the within-cluster sum of squares (inertia), calculated using Equation (1). As K increases, inertia always decreases; with more clusters, points are closer to their respective centroids. Eventually, however, improvements begin to diminish as more clusters are added.



Elbow Method for Choosing K

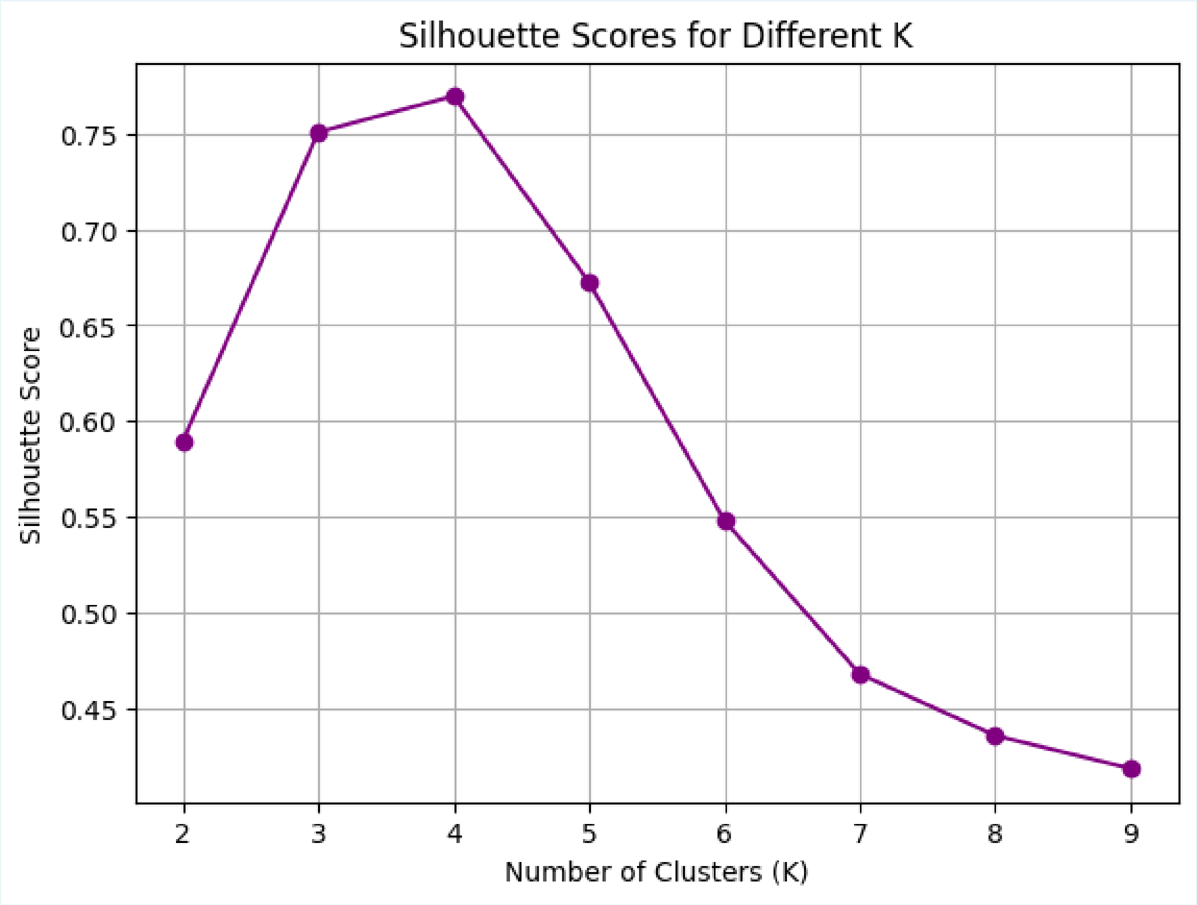
In figure 5, inertia is initially steeply decreasing and then flattens out over time. The point on the plot where it "bends" or "elbows" is a reasonable selection for K, since it represents the best compromise between the complexity of the model and the quality of the fit.

6.2 Silhouette Analysis

The silhouette score also describes how distinct one cluster is from other clusters using silhouette scores to measure the degree to which points in one cluster resemble other points within that same cluster. Each individual point in a dataset has an average distance *a* to all the other points within its own cluster and has an average distance *b* to the closest cluster. The silhouette score of a data point is determined using the formula below:

$$s = (b - a) / max(a, b)$$

Silhouette scores range from -1 to +1, with higher values indicating that a data point belongs more closely to its own cluster than to any of the surrounding clusters, while lower and/or negative scores indicate that that point may be misclassified as being part of another cluster.



Silhouette Scores for Different Values of K

Figure 6 shows that the maximum silhouette score occurs at the "optimal" K value that we will obtain by combining the elbow method and silhouette score analysis to obtain the optimum value for K.

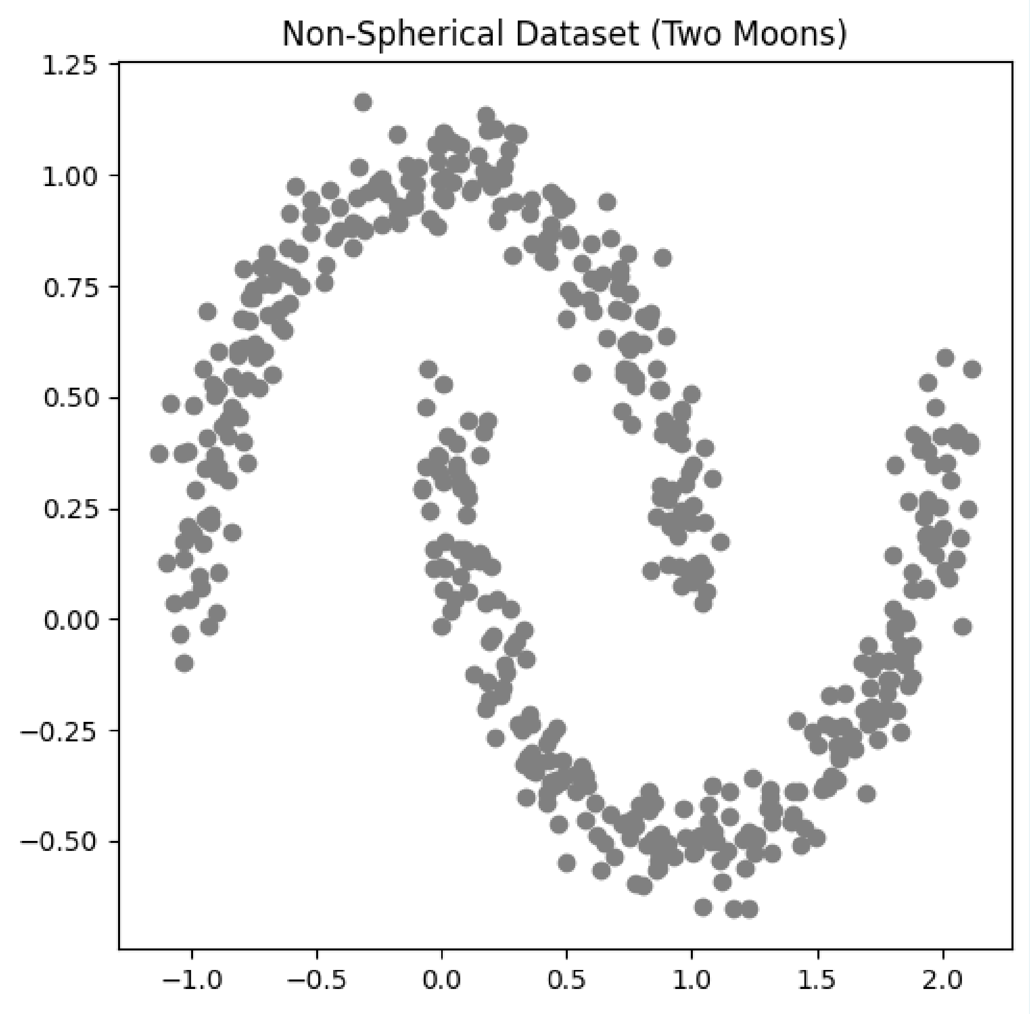
Both of these are generic techniques for model selection: a Generalization Technique for Model Selection that can be applied to many clustering algorithms (not only K-Means).

7. K-Means Limitations and a Failure Example

K-Means is straightforward and useful but not universally applicable. It makes a number of strong assumptions that are sometimes incorrect:

1. That the clusters are compact and approximately spherical when visualized in Euclidean space.
2. That all clusters (or groups of points) have similar size and density.
3. That Euclidean distance is a meaningful way to define cluster membership in the data set being analyzed.

To illustrate these assumptions and examples of when they are not correct, K-Means was applied to the make_moons data set, which consists of two interlocking crescent moon shapes (see Figure 7).



K-Means Failure Case on Non-Spherical Data

As seen in Figure 7, K-Means is attempting to "cut" between the two moons using a straight line. Therefore, many points are found outside of their cluster, even though the two clusters are visually identifiable by a human observer. In this case, the distance from the central point, the centroid, is a poor criterion for determining cluster membership.

Methods such as DBSCAN and Spectral Clustering could be utilized to cluster the make moon data set. This demonstrates that there is no single clustering method that can be applied to all data sets. While K-Means may be a suitable general-purpose clustering method for a particular problem, it may also be inappropriate for use in other situations.

8. Generic Techniques and Models in This Tutorial

This tutorial covers multiple generic techniques:

- **Distance Metrics:** We used Euclidean distance to indicate how similar items are to one another.

- **Iterative Optimisation:** K-Means algorithm has an iterative process for assigning and updating points to clusters.
- **Variance Minimisation:** The objective function seeks to find the minimum within-cluster squared distances.
- **Heuristic Initialisation:** The K-Means++ algorithm builds upon the K-Means algorithm by applying a heuristic method to improve the solution.
- **Model Selection:** The elbow method and silhouette score help to determine the optimal value for K.

Additionally, this tutorial addresses multiple generic models:

- **Centroid-Based Clustering:** Clusters are defined or represented by the means of points in a cluster.
- **Partitioning Models:** The dataset is split into disjoint groups.
- **Unsupervised Learning Models:** The model infers structure without requiring labels on the dataset.

These techniques and models will be relevant throughout the field of Machine Learning and Data Analysis, not only with K-Means.

9. Conclusion

K-Means Clustering, also known as K-means Clustering Algorithm, is one of the simplest and fastest clustering algorithms. It is also one of the most easily interpretable clustering algorithms. It can also help you assess, or evaluate, other clustering algorithms, and it can start your exploration of clustering data.The performance of K-Means Clustering will vary depending on the two user-defined parameters: How you initialize the centroids and what value you select for "K".

In this report, we have:

- Described K-Means Clustering and the ultimate goal of utilising it.
- Demonstrated that random initialisation can lead to less effective clusters or consistencies.
- Illustrated the use of K-Means++ as an effective way of providing better initial conditions and therefore more reliable clustering results.
- Developed an understanding of how centroids are relocated and illustrated the iterative refinement method to initiate the final state of a K-Means model.
- Utilised the elbow and silhouette analysis as generic selection methodologies for K.
- Illustrated the limitations of K-Means on non-spherical data through practical examples.

This understanding can assist practitioners in utilising K-Means more effectively and using it as a valid tool in its area of applicability rather than relying on it as a blanket solution. However, K-Means is an effective and elegant clustering model when using K-Means in conjunction with valid assumptions; otherwise, K-Means provides a starting point for comparison with other more sophisticated methods.

10. References

Arthur, D., & Vassilvitskii, S. (2007). k-means++: The Advantages of Careful Seeding. Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, 1027–1035. Forgy, E. W. (1965). Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classifications. Biometrics, 21(3), 768–769. Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), 100–108. Jain, A. K. (2010). Data Clustering: 50 Years Beyond K-Means. Pattern Recognition Letters, 31(8), 651–666. Lloyd, S. P. (1982). Least Squares Quantization in PCM. IEEE Transactions on Information Theory, 28(2), 129–137. (Originally published as Bell Labs Technical Report, 1957.) MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1, 281–297. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830. Rousseeuw, P. J. (1987). Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Journal of Computational and Applied Mathematics, 20, 53–65. Xu, R., & Wunsch, D. (2005). Survey of Clustering Algorithms. IEEE Transactions on Neural Networks, 16(3), 645–678.