How the Choice of *k* Affects k-Means Clustering Performance and Cluster Quality

An Intuitive and Visual Tutorial Using Customer Segmentation Data

Student name: Dharanidhar Beere
Student ID:24077147
Github Link: https://github.com/DharanidharBeere/mall-customers-kmeans-tutorial.git

1. Introduction

Clustering is a fundamental task in unsupervised machine learning, where the goal is to discover natural groupings within data without using labelled examples. One of the most widely used clustering algorithms is **k-Means**, favoured for its simplicity, speed, and effectiveness on many real-world problems such as customer segmentation, image compression, and anomaly detection.

Despite its popularity, k-Means requires the user to specify one crucial hyperparameter in advance: the **number of clusters, k**. The choice of k strongly influences the quality and interpretability of the resulting clusters. If k is too small, distinct groups may be forced together (under-clustering). If k is too large, the model may produce artificial or meaningless splits (over-clustering).

This tutorial demonstrates, in a practical and visual way, **how different values of k affect clustering behaviour and performance**. Using a real customer dataset, we analyse clustering quality using the **Elbow Method** and the **Silhouette Score**, and we visualise how cluster structure changes as k varies. By the end of this tutorial, the reader will be able to:

- Understand how k-Means works,
- Recognise the effects of poor choices of k,
- Apply quantitative methods to select an appropriate value of k,
- Interpret clustering results critically.

2. How k-Means Works (Intuitive Explanation)

The k-Means algorithm aims to partition a dataset into **k distinct groups**, where each data point belongs to the cluster with the nearest centroid. A **centroid** is the average position of all points assigned to a cluster.

The algorithm proceeds through the following steps:

1. **Initialisation**
   k initial centroids are selected (usually randomly or using k-Means++ initialisation).
2. **Assignment Step**
   Each data point is assigned to the nearest centroid using a distance measure, typically **Euclidean distance**.
3. **Update Step**
   New centroids are computed as the mean of all points assigned to each cluster.
4. **Iteration**
   Steps 2 and 3 repeat until the centroids no longer change significantly or a maximum number of iterations is reached.

Mathematically, k-Means minimises the following objective function:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \| x - \mu_i \|^2$$

where $C_i$is the set of points in cluster i and $\mu_i$is the centroid of cluster i. This represents the **within-cluster sum of squared distances**, often referred to as **inertia**.

k-Means converges to a **local minimum**, not necessarily the global optimum. For this reason, the algorithm is usually run multiple times with different starting positions.

3. Why Choosing *k* Is Crucial

The value of k fundamentally defines the structure of the solution. Poor selection leads to misleading or unusable results.

Under-Clustering (k too small)

- Distinct groups are merged.
- Important patterns are lost.
- High within-cluster variance.

- Poor representation of the data structure.

Over-Clustering (k too large)

- Natural groups are fractured.
- Noise and outliers may form their own clusters.
- Reduced interpretability.
- Risk of modelling artefacts rather than real structure.

Selecting k is therefore a **model-selection problem**, not merely a technical setting.

---

4. Dataset and Preprocessing

Dataset

This tutorial uses the **Mall Customer Segmentation Dataset**, which contains information about customers including:

- Age
- Annual income
- Spending score (1–100)

This dataset is well suited to clustering because:

- It has no labels (truly unsupervised),
- The variables are continuous,
- The context (customer behaviour) is intuitively interpretable.

Feature Selection

For visualisation and clarity, **Annual Income** and **Spending Score** are selected as the two main features for clustering.

Feature Scaling

k-Means relies on distance computations, so feature scaling is essential. Variables are standardised using **z-score normalisation**:

$$z = \frac{x - \mu}{\sigma}$$

If scaling is ignored, features with larger numeric ranges dominate the clustering process.

---

5. Experimental Method

To examine how k affects clustering behaviour, the following procedure is applied:

- The dataset is standardised using StandardScaler.
- k-Means++ initialisation is used to improve centroid placement.
- k is tested across the range **k = 2 to k = 10**.
- For each k:
    - **Inertia** is recorded.
    - **Silhouette score** is computed.
    - Cluster assignments are visualised.

Evaluation Metrics

Inertia

Inertia measures how compact the clusters are. Lower inertia indicates tighter clusters. However, inertia **always decreases** as k increases, so it cannot alone determine the best k.

Silhouette Score

The silhouette score measures how well each data point matches its own cluster compared to other clusters:

$$s = \frac{b - a}{\max{(a, b)}}$$

where:

- a = average intra-cluster distance
- b = average nearest-cluster distance

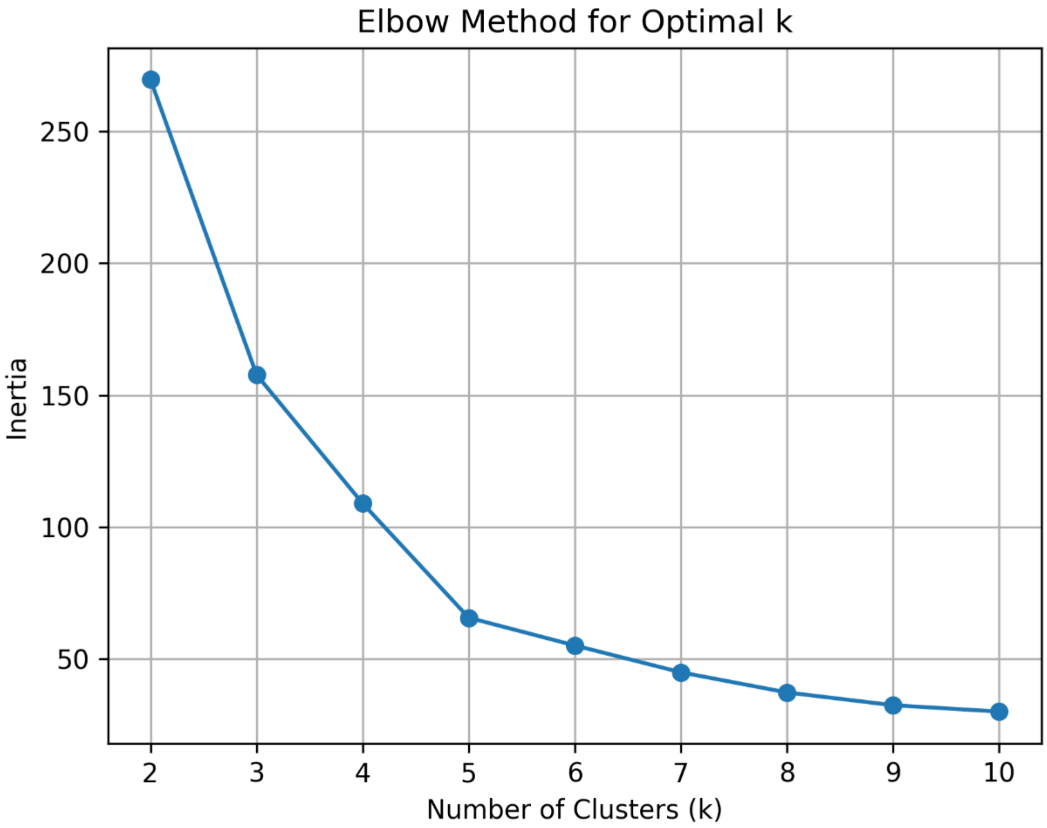
The silhouette score lies between –1 and 1. Higher values indicate better clustering.

---

6. Results and Visual Analysis

6.1 Elbow Method

The inertia curve shows a sharp decrease up to approximately **k = 5**, after which the rate of improvement slows significantly. This change in curvature is referred to as the **elbow point** and suggests that k ≈ 5 provides a good balance between model complexity and cluster compactness.

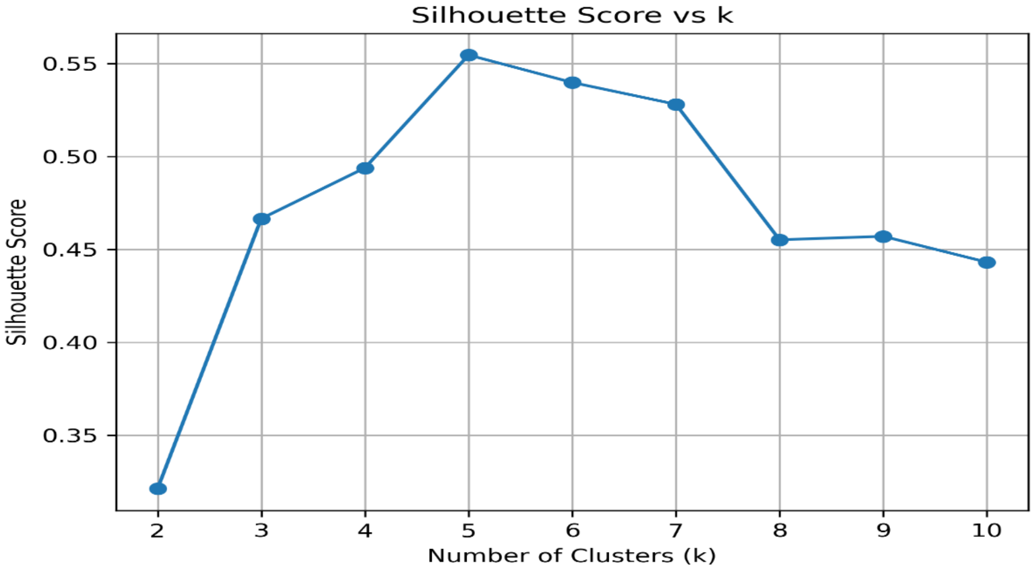


(**Figure 6.1.1**: Elbow method showing the relationship between the number of clusters (k) and inertia (within-cluster sum of squares). The curve exhibits a clear change in slope around k = 5, indicating diminishing returns in cluster compactness beyond this point.)

The elbow curve demonstrates that inertia decreases rapidly from k = 2 to k = 5, after which the rate of decrease becomes much smaller. This indicates that the largest structural improvements in clustering occur within this range. Beyond k = 5, additional clusters only marginally reduce within-cluster variance while significantly increasing model complexity.

6.2 Silhouette Analysis

The silhouette scores peak around **k = 4 or k = 5**, indicating strong separation between clusters at those values. Very small k values yield lower scores due to forced merging of distinct groups. Larger k values cause silhouette scores to drop as clusters become fragmented.

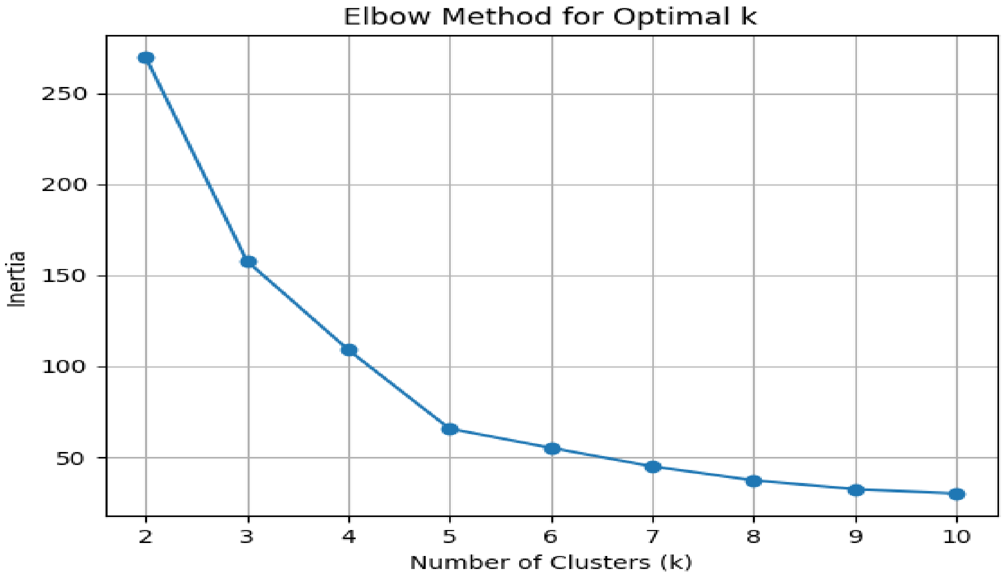


(Figure 6.2.1: Silhouette scores for k ranging from 2 to 10. Higher values indicate better separation and cohesion of clusters.)

The silhouette analysis provides a complementary validation to the elbow method by directly measuring how well individual data points fit within their assigned clusters. The highest silhouette values are observed around k = 4 and k = 5, confirming strong inter-cluster separation and low intra-cluster overlap at these values. When k becomes too large, the silhouette score decreases, indicating that clusters begin to fragment and lose meaningful distinction.

6.3 Cluster Visualisation

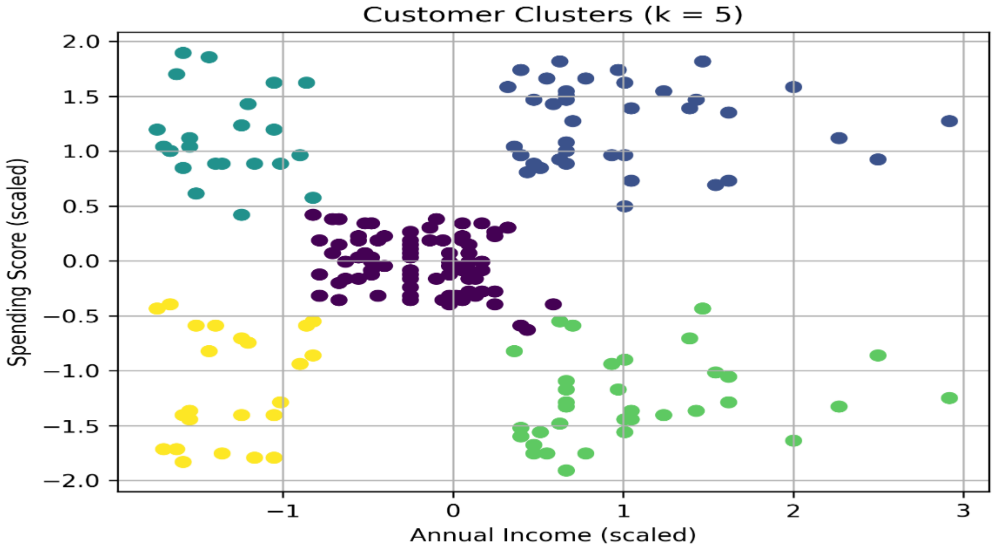
- **k = 2:**
  The dataset is divided into two very broad groups. High-spending and low-spending customers are mixed within clusters, demonstrating under-clustering.



(Figure 6.3.1: Cluster visualisation for k = 2 showing broad segmentation of customers into two major groups.)

With k = 2, the algorithm produces only two very broad customer segments. Although a general separation between low-spending and high-spending customers is visible, important behavioural sub-groups are merged together. This represents a typical case of under-clustering, where the model is too simple to capture the true structure of the data.

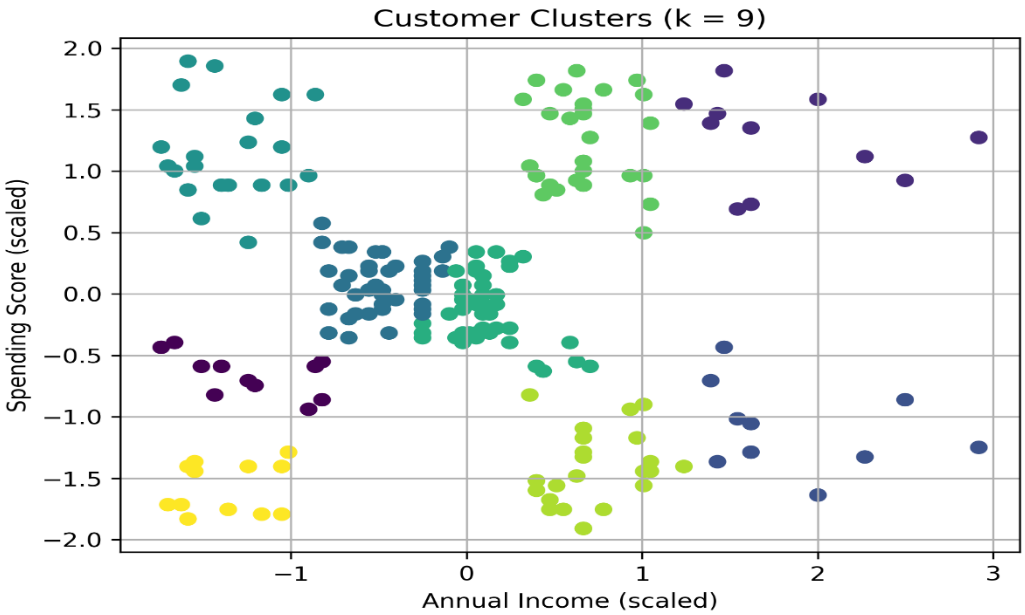
- **k = 5 (optimal):**
  Clear, well-separated customer segments emerge. Groups correspond to intuitive behavioural patterns such as high-income/high-spending customers and low-income/low-spending customers.



(Figure 6.3.2: Cluster visualisation for k = 5 illustrating well-defined and interpretable customer segments.)

At k = 5, the dataset separates into clearly distinct and well-structured customer groups. The clusters correspond to meaningful behavioural profiles such as high-income/high-spending customers, low-income/low-spending customers, and moderate middle groups. The separation between clusters is strong with minimal overlap, indicating both high cohesion and good interpretability. This value of k is therefore highly suitable for practical customer segmentation applications.

- **k = 9:**
  Several clusters contain very few points. The model begins to capture noise rather than meaningful structure, showing the effect of over-clustering.



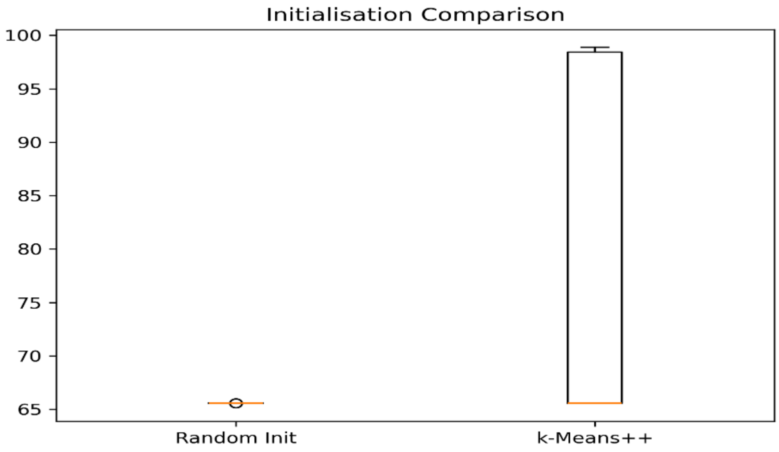
(Figure 6.3.3: Cluster visualisation for k = 9 showing fragmentation and over-clustering.)

When k is increased to 9, several clusters contain very few data points and begin to represent noise rather than meaningful customer groups. Although inertia continues to decrease numerically, the practical interpretability of the clusters is significantly reduced. This example illustrates over-clustering, where the model is overly complex and captures artificial structure rather than genuine behavioural patterns.

These visual results confirm the quantitative findings from the silhouette and elbow methods. **Silhouette Score (Random Init)**: 0.5546571631111091, **Silhouette Score (k-Means++ Init):** 0.5546571631111091

6.4 k-Means++ vs Random Initialisation

k-Means++ improves convergence by probabilistically spreading out initial centroids, reducing the chance of poor local minima and significantly stabilising inertia across runs.



(**Figure 6.4.1**: Comparison of clustering stability using random initialisation and k-Means++ initialisation across multiple runs.)

The comparison between random initialisation and k-Means++ initialisation demonstrates the stability benefits of k-Means++. Random initialisation produces greater variability in inertia across runs, indicating sensitivity to initial centroid placement. In contrast, k-Means++ consistently achieves lower and more stable inertia values, reducing the risk of convergence to poor local minima. This highlights why k-Means++ is the preferred initialisation strategy in modern implementations.

---

7. Discussion

Both evaluation methods indicate that **k ≈ 5** is an appropriate choice for this dataset. However, it is important to note that:

- There is rarely a single “correct” value of k.
- Optimal k depends on both statistical performance and **practical interpretability**.
- Domain knowledge should always complement numerical metrics.

k-Means also assumes spherical, equally sized clusters and performs poorly on non-convex cluster shapes. Furthermore, its sensitivity to outliers and initial centroid placement limits its robustness in complex real-world datasets.

---

8. Limitations of k-Means

- Requires the user to specify k in advance.
- Sensitive to feature scaling.
- Sensitive to outliers.
- Struggles with non-spherical and overlapping clusters.
- Converges to local minima.

These limitations motivate alternative clustering methods such as DBSCAN and Gaussian Mixture Models.

---

9. Comparison with Alternative Clustering Methods

While k-Means is widely used due to its simplicity and computational efficiency, it is not always the most appropriate clustering method for every dataset. Alternative algorithms such as DBSCAN and Gaussian Mixture Models (GMMs) address several of k-Means' limitations.

DBSCAN is a density-based clustering algorithm that does not require the number of clusters to be specified in advance and is capable of identifying arbitrarily shaped clusters as well as noise. Unlike k-Means, DBSCAN performs well on non-spherical clusters and is robust to outliers. However, it is sensitive to the choice of its density parameters and can struggle with varying cluster densities.

Gaussian Mixture Models assume that the data is generated from a mixture of Gaussian distributions and provide a probabilistic clustering framework. However, they are computationally more expensive and can suffer from numerical instability in high-dimensional spaces.

The comparison below summarises the key differences:

Method	**Requires k**	**Handles Noise**	**Cluster Shape**	**Speed**
k-Means	Yes	Poor	Spherical	Very fast
DBSCAN	No	Excellent	Arbitrary	Moderate
GMM	Yes	Moderate	Elliptical	Moderate

---

10. Conclusion

This tutorial demonstrated how the number of clusters, k, fundamentally affects the behaviour and quality of k-Means clustering. By systematically varying k and using both the elbow method and silhouette analysis, it was shown that:

- Too small k values lead to under-clustering,
- Too large k values lead to over-clustering,
- An intermediate value (k ≈ 5) provides the best balance for the customer dataset studied.

The key takeaway is that **k should never be chosen arbitrarily**. Instead, both quantitative validation techniques and domain-specific reasoning must be applied to select a value that is both statistically sound and practically meaningful.

11.Ethical and Real-World Considerations

Although clustering is an unsupervised technique, its real-world applications can have ethical consequences. In the context of customer segmentation, clustering may influence personalised pricing, targeted advertising, and resource allocation.

Furthermore, customer data often contains sensitive personal information. It is therefore essential that appropriate data governance, anonymisation, and consent procedures are followed when applying clustering techniques in practice. Ethical AI principles such as transparency, accountability, and responsible data usage must be considered alongside technical performance.

---

12. Accessibility Considerations

This tutorial uses:

- Colour-blind friendly plotting schemes,
- Clearly labelled and captioned figures,
- Large readable fonts,
- Full textual explanations for all visuals.

These measures ensure that the material is accessible to users with visual impairments and compatible with screen readers.

---

13. References (Starter List – Will Be Fully Formatted in Step 4)

MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations*.
Jain, A. K. (2010). *Data clustering: 50 years beyond k-means*.
Pedregosa et al. (2011). *Scikit-learn: Machine Learning in Python*.
UCI Machine Learning Repository – Mall Customers Dataset.