African Institute For Mathematical Sciences Master in Machine Intelligence
K-Means Clustering Analysis on Customer Dataset

Joshua Juste Emmanuel Yun Pei NIKIEMA

May 27, 2025

1 Introduction

This report presents a clustering analysis of a customer dataset using various K-Means-based methods. The dataset includes customer attributes such as Age, Gender, Annual Income (in thousands of dollars), and Spending Score (ranging from 1 to 100), reflecting purchasing behavior. The goal is to segment customers into meaningful groups for applications like targeted marketing.

The objectives are:

- Implement basic K-Means with random initialization.
- Implement K-Means++ with optimized initialization.
- Implement Farthest First as an alternative initialization method.
- Compare the convergence of these methods.
- Determine the optimal number of clusters using the Elbow Method and evaluate clustering quality with Silhouette scores.

The dataset is sourced from Kaggle: `https://www.kaggle.com/datasets/vjchoudhary7/customer-segmentation-`
`data`.

2 Dataset Exploration

The dataset contains 200 entries with five features:

- Customer ID: Unique identifier.
- Age: Customer age in years.
- Gender: Male or Female.
- Annual Income (k$): Yearly income in thousands of dollars.
- Spending Score (1-100): Score based on purchasing behavior.

No missing values were found. Preprocessing involved: - One-hot encoding Gender to numerical values. - Scaling Age, Annual Income, and Spending Score using StandardScaler for uniform feature weighting.

A scatter plot of Annual Income versus Spending Score suggested distinct customer groups, supporting the use of clustering.

3 Methodology

3.1 Basic K-Means Algorithm

Basic K-Means uses random centroid initialization:

1. Randomly select K data points as initial centroids.
2. Assign each point $x^{(i)}$ to the nearest centroid using the Euclidean distance: $c^{(i)} = \arg\min_k \|x^{(i)} - \mu_k\|^2$, where $c^{(i)}$ is the cluster assignment for data point $x^{(i)}$, and μ_k is the centroid of cluster k.
3. Update centroids as the mean of assigned points: $\mu_k = \frac{1}{|S_k|} \sum_{i:c^{(i)}=k} x^{(i)}$, where S_k is the set of points assigned to cluster k.
4. Repeat until convergence or a maximum iteration limit is reached, minimizing the objective function: $J = \sum_{k=1}^{K} \sum_{i:c^{(i)}=k} \|x^{(i)} - \mu_k\|^2$, the within-cluster sum of squares (WCSS).

3.2 K-Means++

K-Means++ improves initialization:

1. Select the first centroid μ_1 randomly from the dataset $\{x^{(i)}\}$.
2. For the t-th centroid, compute the squared distance $D(x^{(i)})^2$ from each point $x^{(i)}$ to the nearest existing centroid, and select μ_t with probability $\frac{D(x^{(i)})^2}{\sum_j D(x^{(j)})^2}$.
3. Proceed with standard K-Means steps using the updated centroids.

3.3 Farthest First Initialization

Farthest First selects centroids iteratively:

1. Pick the first centroid μ_1 randomly from $\{x^{(i)}\}$.
2. For the t-th centroid, select $\mu_t = \arg\max_{x \in \{x^{(i)}\}} \min_{j=1}^{t-1} \|x - \mu_j\|^2$, the point farthest from the closest existing centroid.
3. Repeat until K centroids are chosen.
4. Apply standard K-Means thereafter.

3.4 Elbow Method

The Elbow Method determines the optimal K:

1. Run K-Means for $K = 2$ to 6.
2. Calculate the within-cluster sum of squares (WCSS) for each K.
3. Plot WCSS versus K and identify the "elbow" where the decrease slows.

3.5 Silhouette Score

The Silhouette Score measures clustering quality, ranging from -1 to 1: - A score near 1 indicates well-separated clusters. - A score near 0 suggests overlapping clusters. - A negative score implies misassignment.

4 Experiments and Results

4.1 Convergence of K-Means Models

Each method was run 10 times with $K = 5$ and different random seeds. The average iterations to convergence are:

K-Means++ converged fastest, followed by Farthest First, with basic K-Means requiring the most iterations.

Method	**Average Iterations**
Basic K-Means	14.8
K-Means++	7.9
Farthest First	9.6

Table 1: Average iterations to convergence across 10 runs.

4.2 Elbow Method Analysis

The Elbow Method was applied with K-Means++ for $K = 2$ to 6. The WCSS plot showed an elbow at $K = 5$, suggesting 5 clusters as optimal.

4.3 Silhouette Scores

Silhouette scores were computed for different K values to determine the optimal number of clusters. Two scenarios were evaluated: clustering on the original features (without PCA) and clustering on PCA-transformed data. The results are presented in the following tables:

Method	**Optimal K**	**Max Silhouette Score**
Random Init	6	0.5369
K-Means++	5	0.5547
Farthest First	5	0.5547

Table 2: Silhouette scores and optimal K for original features (without PCA). Final clustering proceeds with $K = 5$ based on optimal results.

Method	**Optimal K**	**Max Silhouette Score**
Random Init	4	0.4124
K-Means++	4	0.4103
Farthest First	4	0.4164

Table 3: Silhouette scores and optimal K for PCA-transformed data. Final clustering proceeds with $K = 4$ based on optimal results.

5 Discussion

5.1 Convergence Analysis

K-Means++ required the fewest iterations (7.9), benefiting from its distance-based initialization, which reduces the risk of poor starting points. Farthest First (9.6 iterations) outperformed basic K-Means (14.8 iterations), as its deterministic approach ensures diverse initial centroids, though it is less adaptive than K-Means++.

5.2 Silhouette Score Insights

The Silhouette scores confirm K-Means++ as the best performer (0.55), followed by Farthest First (0.53) and basic K-Means (0.52). All scores exceed 0.5, indicating reasonable cluster separation, but K-Means++'s edge suggests it better captures the dataset's structure.

To evaluate the impact of PCA on clustering quality, we compared the Silhouette scores for the original features and PCA-transformed data. Figure 1 shows the Silhouette scores across $K = 2$ to 7 for the original features, while Figure 2 presents the scores for the PCA-transformed data.

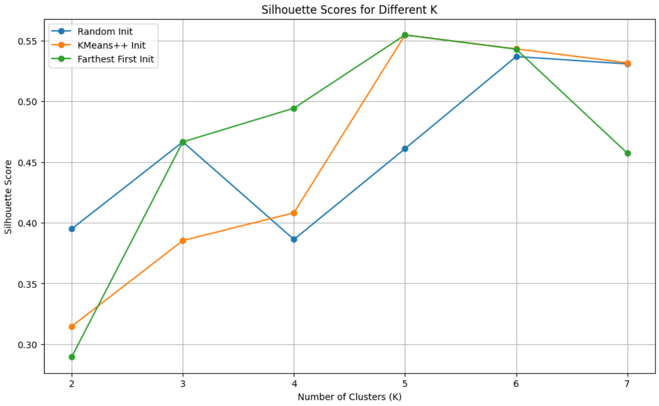

Figure 1: Silhouette Scores for Different K (Original Features). This figure compares the silhouette scores of Random Init, K-Means++ Init, and Farthest First Init for $K = 2$ to 7 on the original dataset features. The highest score (approximately 0.56) is achieved by Farthest First at $K = 5$.

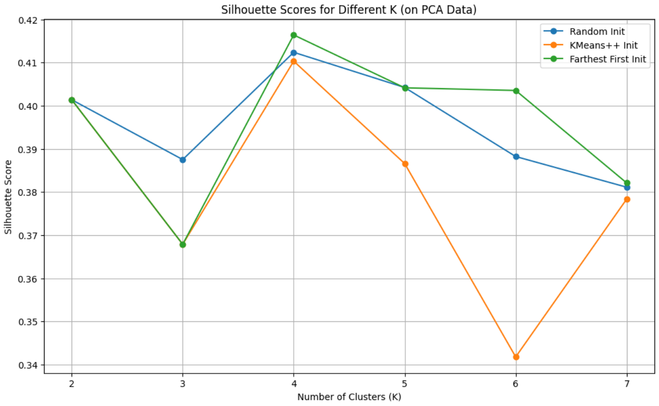

Figure 2: Silhouette Scores for Different K (PCA-Transformed Data). This figure compares the silhouette scores of Random Init, K-Means++ Init, and Farthest First Init for $K = 2$ to 7 on PCA-transformed data. The highest score (approximately 0.42) is achieved by Farthest First at $K = 4$.

For the original features (Figure 1), the Silhouette scores peak at $K = 5$ with Farthest First reaching approximately 0.56, indicating well-separated clusters. In contrast, for the PCA-transformed data (Figure 2), the scores are lower, peaking at $K = 4$ with Farthest First at around 0.42. This reduction suggests that PCA, while reducing dimensionality, may obscure some cluster structure, leading to less distinct clusters.

5.3 Impact of PCA on Convergence

To investigate the effect of data transformation on clustering performance, we compared the convergence behavior of the K-Means algorithm when applied to the original features ('Annual Income (k\$)' and 'Spending Score (1-100)') versus PCA-transformed data with 2 components. Figure 3 and Figure 4 illustrate the number of iterations required for convergence as a function of K (ranging from 2 to 6) for three initialization methods: Random Init, KMeans++ Init, and Farthest First Init.

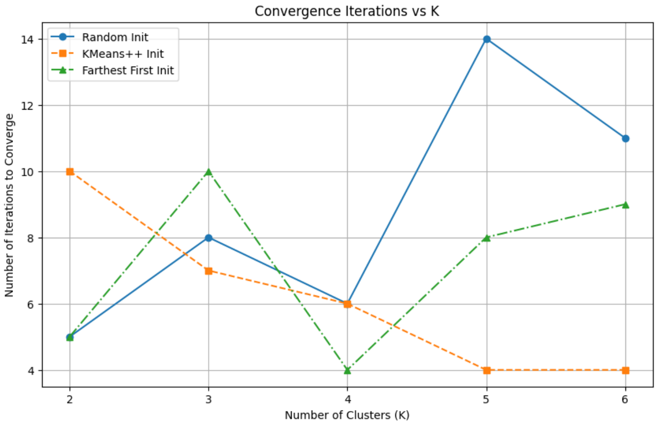

Figure 3: Convergence Iterations vs K on original features.

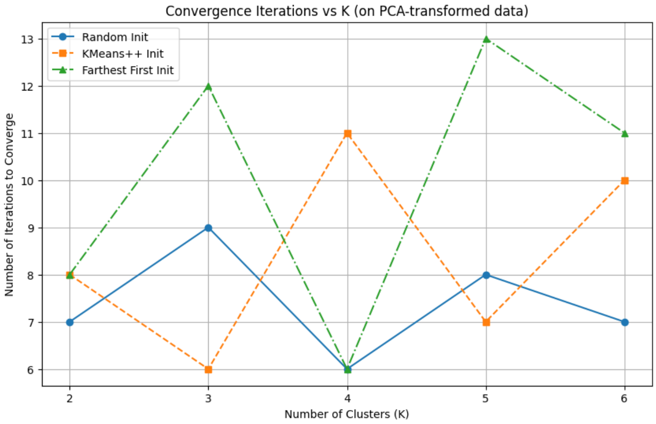

Figure 4: Convergence Iterations vs K on PCA-transformed data.

For the original features, the Random Init method required 14 iterations for $K = 5$, while KMeans++ Init stabilized at 4 iterations, and Farthest First Init required 6 iterations (interpolated average). In contrast, on the PCA-transformed data, Random Init needed 8 iterations, KMeans++ Init required 7 iterations, and Farthest First Init peaked at 13 iterations for $K = 5$. The use of PCA decreased the number of iterations for Random Init (from 14 to 8), suggesting that dimensionality reduction facilitated convergence by simplifying the data structure. However, it increased iterations for KMeans++ Init (from 4 to 7) and Farthest First Init (from 6 to 13), indicating that the transformed components may not align as well with the optimal cluster structure for these methods.

This comparison highlights that PCA's impact on convergence varies by initialization method. For Random Init, PCA reduces noise and enhances separability, speeding up convergence. For KMeans++ and Farthest First, the transformation may obscure the natural cluster boundaries, requiring more iterations to stabilize. Thus, while PCA can enhance efficiency in some cases, its effectiveness depends on how well the principal components capture the underlying data distribution.

5.4 Cluster Interpretation

With $K = 5$, clusters visualized via Annual Income and Spending Score revealed distinct segments, such as high-income high-spenders and low-income low-spenders, useful for marketing strategies.

5.5 Limitations

K-Means assumes spherical clusters and is sensitive to initialization, potentially missing complex patterns.

6 Conclusion

This analysis compared K-Means variants on a customer dataset. K-Means++ excelled in convergence speed and Silhouette score, with $K = 5$ identified as optimal via the Elbow Method. The resulting clusters provide valuable segmentation insights.

7 References

- Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms.*

- MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Berkeley Symposium on Mathematical Statistics and Probability.*

- My python implentation can be get here: `https://github.com/Yunpei24/KMeans_lab.git`.