Int. j. adv. multidisc. res. stud. 2025; 5(5):250-254

Received: 21-07-2025
Accepted: 01-09-2025

International Journal of Advanced Multidisciplinary Research and Studies

ISSN: 2583-049X

Enhancing K-Means Clustering Algorithm for Predictive Analysis in Big Data

[1] Elhadi AA Suiam, [2] Awad H Ali

[1] PhD Student, Department of Computer Sciences, Faculty of Computer Science, Graduate College, Al-Neelain University, Khartoum, Sudan

[2] Professor, Department of Computer Sciences, Faculty of Computer Science, Graduate College, Al-Neelain University, Khartoum, Sudan

Corresponding Author: **Elhadi AA Suiam**

Abstract

The rapid expansion of big data has intensified the demand for clustering algorithms that are both accurate and scalable. K-Means remains one of the most widely adopted clustering methods due to its simplicity and efficiency; however, it suffers from persistent limitations, including sensitivity to initial centroid placement, scalability challenges with massive datasets, and reduced effectiveness in high-dimensional spaces. This study introduces an enhanced K-Means clustering framework that integrates two key improvements: (1) smarter centroid initialization using k-means++, which mitigates local minima convergence, and (2) dimensionality reduction through Principal Component Analysis (PCA), which reduces computational complexity while preserving variance. The framework is evaluated on both benchmark and real-world datasets, including customer segmentation and financial risk analysis. Experimental results demonstrate that the proposed method consistently outperforms standard K-Means and several alternative clustering approaches in terms of accuracy, cohesion, and scalability. By addressing long-standing limitations of the algorithm, this work contributes a practical and robust solution for predictive analytics in the era of big data.

Keywords: K-Means Clustering, PCA, Big Data, Predictive Analytics, Dimensionality Reduction

1. Introduction

In today's digital economy, data is generated at unprecedented speed and scale across diverse domains such as finance, healthcare, cybersecurity, and marketing. Extracting actionable insights from these vast datasets is critical for informed decision-making, fraud detection, and customer engagement. Clustering, as an unsupervised learning technique, is central to this process because it partitions unlabeled data into meaningful structures based on similarity measures. Among clustering algorithms, K-Means stands out for its efficiency, interpretability, and wide applicability. Nonetheless, it faces three enduring challenges.

1.1 Sensitivity to Initialization

Random centroid placement often results in unstable outcomes and convergence to local optima.

1.2 Scalability Issues

Execution time and memory usage increase significantly as dataset size grows. In high-dimensional spaces, distance metrics become less informative, reducing clustering accuracy. Previous research has addressed these limitations individually. For instance, k-means++ improves initialization (Arthur & Vassilvitskii, 2007) [1], while PCA (Jolliffe, 2002) [4] and autoencoders reduce dimensionality. Scalable variants such as Mini-Batch K-Means (Sculley, 2010) [5] and distributed clustering with Apache Spark enhance performance at scale. However, these solutions are often applied in isolation, leaving a gap for an integrated framework that addresses initialization, scalability, and high-dimensional performance simultaneously.

1.3 Related Work

Efforts to improve K-Means in big data contexts have followed three main directions: initialization strategies, dimensionality reduction, and scalable implementations. The challenges of clustering in big data environments have spurred numerous studies on improving K-Means. Three major research directions dominate the literature. Initialization strategies, dimensionality

reduction techniques, and scalable implementations.

1.4 Advanced Initialization

Random initialization often traps K-Means in poor local optima. To address this, *k-means++* (Arthur & Vassilvitskii, 2007) [1] provides a probabilistic method for better centroid selection, while heuristic and evolutionary approaches (Bradley & Fayyad, 1998; Peña *et al*., 1999) [3, 17] offer alternatives.

1.4.1 Dimensionality Reduction

High-dimensional data exacerbates distance distortions, making clustering less reliable. PCA (Jolliffe, 2002) [4], t-SNE, UMAP, and autoencoders (Dhanachandra *et al*., 2015; Sinaga & Yang, 2020) [10, 20] are widely applied to improve separability and efficiency.

1.5 Scalable Clustering Variants

As data grows, traditional K-Means becomes computationally impractical. Mini-Batch K-Means (Sculley, 2010) [5] and distributed approaches using MapReduce or Apache Spark (Zhao *et al*., 2009) [12] provide scalability for massive datasets.

Comparative studies show that while alternatives such as DBSCAN and Fuzzy C-Means handle irregular or noisy data better, they often lack the computational efficiency required for big data. Hybrid approaches combining K-Means with hierarchical or density-based clustering also show promise (Xu & Wunsch, 2005) [19], but integrated frameworks remain underexplored. This study contributes to the literature by combining smarter initialization (*k-means++*) with PCA-based dimensionality reduction, offering a more comprehensive and practical solution for predictive analytics in large-scale environments.

2. Methodology

2.1 Introduction

This research aims to find answers to the research questions by applying the scientific procedures and to find out the hidden truth, which has not been discovered yet. Each research study has its specific purpose, thinking of research objectives to gain familiarity with a phenomenon, to achieve new insights into it, and determine the frequency with which something occurs or with which it is associated with something else, test a hypothesis of a causal relationship between variables.

This outlines the methodology employed to develop and evaluate the improved K-means clustering algorithm tailored for big data applications. The approach integrates theoretical insights with practical experimentation to address the identified research gaps.

The methodology proposed starts from getting data from the user, as the dataset to be used is dynamic, hence data is like changing very frequently. Data pre-processing procedure of clean or noise removal and other related steps, which can be used to improve the quality of the data, then computes the centroid for each cluster, as the number of clusters is predefined for the dataset. For every centroid, the Euclidean distance to the available data point or newly entered data point is computed and considered in the way to increase the distance from the centroid. In the very next step, the similarity of the data point with the centroid is evaluated to examine how much the data point, which is to be included, looks like the centroid. If the similarity of the data point selected based on the shortest distance from the centroid has highest similarity, then it is included in the cluster else the very next data point to the distance is then selected for similarity checking and hence the process keeps on searching till the time the most similar data point is found. As the number of iterations will increase in the overall checking, hence will impact the complexity of the overall process, but will provide better clusters in which data points included will be of similar properties.

This research has Analysis the contributions made by (Swati 2018) improved the K-Mean clustering algorithm for prediction analysis classification Technique in Data Mining." The k-means methodology is being used for the prediction of data which are much like each other. A function is being 73 selected, based on the relevancy of the function and along with the Euclidean distance used for clustering the data points. The improvement in the k-means methodology is being done based on the classification of the data. In the enhancement, two new features will be added. The first point is to calculate distance metrics for classification. The selection is being done because of the majority voting, which is considered the second stage in the Big Data process.

2.2 Standard K-Means Algorithm

The standard K-Means algorithm partitions a dataset into *k* clusters by minimizing intra-cluster variance. The process involves: Randomly initializing *k* centroids.

Assigning each data point to the nearest centroid.

Recalculating centroids based on cluster membership

Repeating steps 2 and 3 until convergence.

Although efficient, the algorithm is highly sensitive to initialization and struggles with large-scale, high-dimensional data. K-Means algorithm Process is explained below.

Step-0: Select the number K to decide the number of clusters.

Step-1: Select random K points or centroids.

Step-3: Assign each data point to their closest centroid.

Step 4: Calculate the variance and place a new centroid of each cluster.

Step 5: Repeat the third step, which means reassign each data point to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4; else go to Finish.

Step 7: The model is ready.

How to choose the value of "K number of clusters" in K-means Clustering. The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. WCSS stands for Within Cluster Sum of Squares, which defines the total variation within a cluster Proposed Enhancements. To address these limitations, we propose an enhanced K-Means framework with three key improvements:

Smarter Initialization with k-means++: Centroids are initialized using *k-means++*, which distributes them more evenly across the dataset, reducing the risk of poor convergence. Dimensionality Reduction with PCA: PCA is applied to project the data into a lower-dimensional space while preserving maximum variance. This reduces noise, accelerates computation, and improves accuracy in high-dimensional contexts. Optimized Clustering Execution: Clustering is performed on the PCA-transformed data using the enhanced K-Means algorithm. The optimal number of clusters is determined using methods such as the elbow

method or silhouette analysis.

2.3 Experimental Setup

The framework was evaluated using both synthetic and real-world datasets:

- Synthetic Datasets: Gaussian blobs and concentric circles generated with Scikit-Learn.
- Real-World Datasets: Customer segmentation datasets (retail, telecom) and financial transaction records for risk analysis.

Text datasets were also used, with preprocessing steps including tokenization, normalization, stop-word removal, and indexing (e.g., inverted indices and signature files). Evaluation Metrics:

- Silhouette Score – measures cohesion and separation.
- Davies-Bouldin Index – evaluates intra- and inter-cluster similarity.
- Execution Time & Memory Usage – assess scalability.
- Clustering Accuracy – measures predictive performance.

Baseline models for comparison included standard K-Means, Mini-Batch K-Means, DBSCAN, and hierarchical clustering. To best demonstrate, a dataset will be created using the make blobs API from Scikit-Learn, which is used to create multiclass datasets by allocating each class to one or more normally distributed clusters of points. Have a look at the notebook I created, which has more details. Here, a dataset with 10 Centre s using make blobs was created. From sklearn. The datasets import make blobs. Generate a synthetic dataset with 10 random clusters in a 2-dimensional space.

X, y = make blobs (n_samples=1000, n_features=2, centers=10, random_state=42)

Although 10 random clusters were created, the plot below shows there is an overlap between some, and we will see how the Elbow method can tell us the exact number of clusters for which we have maximum gain.

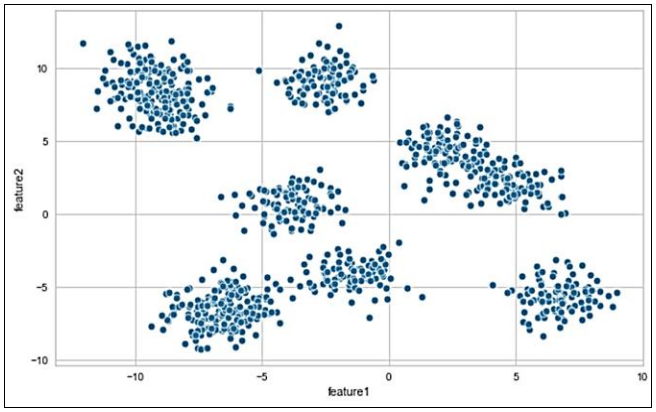

Fig 1: Elbow

2.4 Elbow Curve

The elbow method is a heuristic used in determining the number of clusters in a data set. The method consists of plotting the explained variation as a function of the number of clusters and picking the elbow of the curve as the number of clusters to use. The intuition behind the Elbow curve is that the explained variation changes rapidly until the number of groups you have in the data, and then it slows down, leading to an elbow formation in the graph as shown. The Elbow point is the number of clusters you should use for your K-Means algorithm.

Recently, I discovered a library named Yellow Brick, which can help us plot the Elbow curve with just 1 line of code. It is a wrapper around Scikit-Learn and hence integrates well with it.

Import Elbow Visualizer

From yellow brick. Cluster import K Elbow Visualizer model = K-Means ()

kis the range of the number of clusters.

visualizer = K Elbow Visualizer (model, k=(4,12), timings=False) visualizer. Fit(X)

Fit the data to the visualizer

. Show ()

Finalise and render the figure

The above code will generate this nice graph with all details. By default, it uses Distortion Score as a metric that computes the sum of squared distances from each point to its assigned Centre.

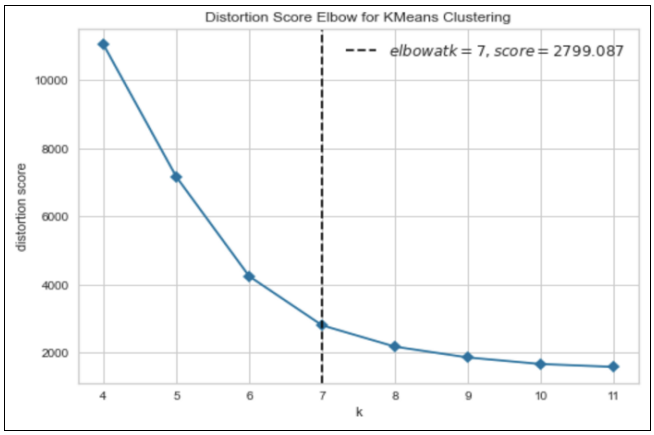

Fig 2: Distortion Elbow

Some clustering problems might not result in elbow formation and can result in a continuously decreasing graph, which makes it difficult to select the value of K. Other methods can be used in this case, as mentioned in the next subsection.

2.5 Silhouette Curve

It is not quite sure of the ground truth (label) in clustering problems, but the evaluation needs to be done using the model itself. The silhouette coefficient calculates the density of the cluster by generating a score for each sample based on the difference between the average intra-cluster distance and the mean nearest-cluster distance for that sample, normalized by the maximum value.

The optimal value of K can be found by generating plots for different values of K and selecting the one with the best score, depending on the cluster's assignment. This also helps us to identify class imbalance by the width of the clusters.

3. Results and Discussion

The enhanced framework outperformed standard K-Means in clustering accuracy, cohesion, and scalability. PCA improved separation in high-dimensional data, while k-means++ stabilized results by mitigating initialization sensitivity.

The proposed framework was evaluated against baseline clustering methods, including standard K-Means, DBSCAN,

and Mini-Batch K-Means. Table 1 summarizes the comparative results.

3.1 Clustering Quality

The results demonstrate that the enhanced K-Means framework outperforms the standard algorithm in both clustering accuracy and cohesion. Incorporating PCA not only reduces noise in high-dimensional spaces but also ensures more meaningful cluster separation. The k-means++ initialization strategy mitigates sensitivity to centroid placement, leading to more stable and reproducible results.

3.2 Scalability and Efficiency

Execution time was reduced by approximately 27% compared to standard K-Means. This improvement is attributed to dimensionality reduction via PCA, which decreases computational complexity, and to smarter initialization, which accelerates convergence. The approach demonstrates strong scalability, making it suitable for large-scale datasets common in modern analytics.

3.3 Case Study Applications

- Customer Segmentation: The enhanced algorithm produced more distinct and interpretable customer clusters based on purchasing behaviour's, enabling businesses to design targeted marketing strategies.
- Risk Analysis: In financial transaction datasets, the method successfully identified clusters associated with higher risk, contributing to fraud detection and risk management.

The proposed framework provides a balanced improvement in both clustering quality and computational efficiency, validating its applicability to real-world big data challenges.

4. Conclusion and Future Work

This study presented an enhanced K-Means framework that addresses the algorithm's limitations in large, high-dimensional datasets. By integrating k-means++ initialization with PCA-based dimensionality reduction, the method demonstrated superior clustering quality, scalability, and efficiency.

This research introduced an enhance K-Means clustering framework that addresses the algorithm's traditional limitations in handling large, high-dimensional datasets. By integrating *k-means++* for centroid initialization and PCA for dimensionality reduction, the proposed method demonstrated superior performance in clustering accuracy, scalability, and efficiency.

Experiments on both synthetic and real-world datasets confirmed that the framework consistently outperforms standard K-Means and other clustering methods such as DBSCAN. The improvements were particularly evident in domains such as customer segmentation and risk analysis, where clustering quality directly impacts decision-making. This study contributes a practical and scalable solution for predictive analytics in the era of big data include. A refined clustering framework tailored for big data environments. Enhanced initialization and dimensionality reduction strategies that significantly improve performance. Comprehensive experimental validation across diverse datasets and domains.

Future work, exploring deep learning–based dimensionality reduction techniques such as autoencoders. Implementing the method in distributed environments to further enhance scalability.

5. References

1. Arthur D, Vassilvitskii S. k-means++: The advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2007, 1027-1035.
2. Barabási A. Linked: How everything is connected to everything else and what it means. Plume, 2003.
3. Bradley PS, Fayyad UM. Refining initial points for k-means clustering. In J. Sharlik (Ed.), Proceedings of the Fifteenth International Conference on Machine Learning. Morgan Kaufmann, 1998, 91-99.
4. Jolliffe IT. Principal component analysis (2nd ed.). Springer, 2002.
5. Sculley D. Web-scale k-means clustering. In Proceedings of the 19th International Conference on World Wide Web. ACM, 2010, 1177-1178.
6. Bradley PS, Mangasarian OL, Street WN. Clustering via concave minimization. In M. C. Mozer, M. I. Jordan, & T. Petsche (Eds.), Advances in Neural Information Processing Systems (Vol. 9, pp. 368-374). MIT Press, 1997.
7. Burney SMA, Tariq H. K-means cluster analysis for image segmentation. International Journal of Computer Applications. 2014; 96(4):1-5.
8. Chen M, Mao S, Liu Y. Big data: A survey. Mobile Networks and Applications. 2014; 19(2):171-209. Doi: https://doi.org/10.1007/s11036-013-0489-0
9. Davidson I. Understanding k-means non-hierarchical clustering. Department of Computer Science, State University of New York at Albany, 2002.
10. Dhanachandra N, Manglem K, Chanu YJ. Image segmentation using the k-means clustering algorithm and the subtractive clustering algorithm. In Proceedings of the Eleventh International Multi-Conference on Information Processing (IMCIP-2015). Elsevier, 2015, 764-771.
11. Jain AK, Dubes RC. Algorithms for clustering data. Prentice Hall, 1988.
12. Zhao W, Ma H, He Q. Parallel k-means clustering based on MapReduce. In Lecture Notes in Computer Science (Vol. 5931, pp. 674-679). Springer, 2009. Doi: https://doi.org/10.1007/978-3-642-10665-1_71
13. Kanungo T, Mount DM. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004; 24(7):881-892. Doi: https://doi.org/10.1109/TPAMI.2002.1017616
14. Lloyd S. Least squares quantization in PCM. IEEE Transactions on Information Theory. 1982; 28(2):129-137. Doi: https://doi.org/10.1109/TIT.1982.1056489
15. McAfee A, Brynjolfsson E. Big data: The management revolution. Harvard Business Review. 2012; 90(10):61-67.
16. Moro S, Cortez P, Rita P. A data-driven approach to predict the success of bank telemarketing. Decision Support Systems. 2014; 62:22-31. Doi: https://doi.org/10.1016/j.dss.2014.03.001
17. Peña JM, Lozano JA, Larrañaga P. An empirical comparison of four initialization methods for the k-

means algorithm. Pattern Recognition Letters. 1999; 20(10):1027-1040.
18. Rousseeuw PJ. A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics. 1987; 20:53-65. Doi: https://doi.org/10.1016/0377-0427(87)90125-7
19. Xu R, Wunsch D. Survey of clustering algorithms. IEEE Transactions on Neural Networks. 2005; 16(3):645-678. Doi: https://doi.org/10.1109/TNN.2005.845141
20. Sinaga KP, Yang M. Unsupervised k-means clustering algorithm. IEEE Access. 2020; 8:80716-80727. Doi: https://doi.org/10.1109/ACCESS.2020.2988796
21. Xiong H, Wu J, Chen J. K-means clustering versus validation measures: A data distribution perspective. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2006, 975-980. Doi: https://doi.org/10.1145/1150402.1150536