مجلة جامعة بني وليد للعلوم الإنسانية والتطبيقية
Bani Waleed University Journal of Humanities and Applied Sciences
تصدر عن– جامعة بني وليد – ليبيا
Website: https://jhas-bwu.com/index.php/bwjhas/index
المؤتمر الدولي الرابع للعلوم و الهندسة
المجلد الحادي عشر – العدد الثاني – 2026 عدد خاص الصفحات (91 -103)
ISSN 3005-3900

Unsupervised Learning Using K-means algorithm

Abdulwahed F. Almarimi [1]*, Ali M. Jellah [2], Doha A. Alghannai [3]
[1] Department of Computer Science, Faculty of Education , Bani Waleed University, Bani Walid, Libya.
[2.3] Department of Computer Science, Faculty of Information Technology , Bani Waleed University, , Bani Walid, Libya
abdulwahed.almarimi@bwu.edu.ly

التعلم غير خاضع للإشراف باستخدام خوارزمية K-means

عبد الواحد فرج المريمي*[1]، علي محمد جلاح [2] ، ضحى عبد الرحمن الغناي[3]
[1] قسم علوم الحاسوب، كلية التربية ، جامعة بني وليد، بني وليد، ليبيا.
[3.2] قسم علوم الحاسوب، كلية تقنية المعلومات، جامعة بني وليد، بني وليد، ليبيا.

Received: 13-11-2025	Accepted: 19-12-2025	Published: 01-03-2026

الملخص:

يهدف هذا البحث إلى تطبيق خوارزمية k-means، التي تُعتبر إحدى أهم تقنيات التعلم غير المُشرف عليه في تجميع البيانات غير المصنفة. وقد اختبرنا تطبيقها على مجموعة بيانات ثنائية الأبعاد تتكون من (300،2) مأخوذة من منصة Kaggle. قمنا بتنزيل البيانات ثم حددنا يدويًا عدد المجموعات K=3، حيث إن تحديد المجموعات هو المشكلة الرئيسية في الخوارزمية. كما حددنا عدد التكرارات T=6، وأظهرت النتائج أن الخوارزمية تحسنت تدريجيًا عبر التكرارات. استخدمنا مقاييس التقييم لتقييم أداء الخوارزمية، حيث استخدمنا دالة الهدف، التي انخفضت من 4719.65 إلى 266.65 عند الاستقرار النهائي. كما أظهرت مقاييس التماسك انخفاضًا ملحوظًا، مما يعكس الترابط بين النقاط داخل كل مجموعة. أما مقياس التقييم «الفصل» فيُظهر المسافة بين المجموعات. تشير هذه النتائج إلى فعالية الخوارزمية في تقسيم البيانات إلى مجموعات في وقت قصير وبكفاءة عالية. ومع ذلك، فإن الاعتماد على الإدخال اليدوي لعدد المجموعات K يمثل مشكلة رئيسية للخوارزمية ويتطلب المزيد من الحلول. لذلك، ينبغي أن تستكشف الأبحاث المستقبلية طرقًا لحل هذه المشكلة، خاصة في حالة مجموعات البيانات الكبيرة، مثل استخدام طريقة (Elbow)، حيث إن الجمع بين هذه الطرق يعزز النتائج ويحدد k بطريقة غير يدوية، مما يجعل عملية التجميع أكثر دقة وفعالية.

الكلمات الدالة: التجميع، خوارزمية K-Means، التعلم الآلي، التعلم غير المُشرف عليه.

Abstract

This research aims to apply the k-means algorithm, which is considered one of the most important unsupervised learning techniques in clustering unlabeled data. We tested its application on a two-dimensional dataset consisting of (300,2) taken from the Kaggle platform. We downloaded the data and then manually specified the number of clusters K=3, as specifying the clusters is the main problem in the algorithm. We also specified the number of iterations T=6, and the results showed that the algorithm gradually improved across iterations. We used evaluation metrics to assess the performance of the algorithm, where we used the objective function, which decreased from 4719.65 to 266.65 at final stability. The Cohesion metrics also showed a significant decrease, reflecting that the points are interconnected within each cluster. The Separation evaluation metric shows the distance between clusters. These results indicate the effectiveness of the algorithm in dividing data into clusters in a short time and with high efficiency. However, relying on manually entering the number of clusters K is a major problem for the algorithm and requires further solutions. Therefore, future work should explore methods to solve this problem, especially in the case of large data sets, such as using the Elbow method, as combining such methods enhances the results and selects k in a non-manual way, making the clustering process more accurate and effective. aggregation process more accurate and effective.

Keywords*:* Clustering, K-Means Algorithm, Machine learning, Unsupervised Learning.

Introduction

Recent decades have seen growing interest in machine learning algorithms, particularly unsupervised learning, which is considered one of the fundamental pillars of data analysis, where the system is trained without any labeled data. In other words, the algorithm learns identify patterns and relationships within the data on its own, without any prior knowledge or guidance. This feature makes it a powerful tool for data analysis, especially in cases where labeled data is either unavailable or too expensive to obtain. In this research paper, the K-Means Algorithm strategy is explained in simple terms. It is based on unsupervised learning, and clustering is one of the most prominent unsupervised learning methods. This technique separates data into different groups and objects, where similar objects are placed in one group, while different objects are placed in different groups [1].The K-Means algorithm is one of the most prominent clustering algorithms, as it divides data into homogeneous groups according to their internal similarity. Despite the simplicity and effectiveness of this algorithm, it faces several challenges when applied , such as reliance on a random initial selection of centers (Initial Centroids), and the difficulty of determining the optimal number of clusters K in advance. If the number is inappropriate, the results may affect the accuracy of the clustering. Several recent studies have pointed to the importance of addressing these limitations. Naeem et al. (2023) provided a comprehensive review of unsupervised learning algorithms, highlighting their importance in analyzing unlabeled data and their applications in multiple fields such as computer vision and natural language processing. They emphasized the position of K-Means among the most widely used clustering algorithms [2]. Kowsic et al. (2024) proposed an improved version of K-means based on combining the elbow method with computational improvements to automatically determine the number of clusters and reduce time complexity. The results showed that the developed algorithm is more efficient than traditional K-means, especially when dealing with large or high-dimensional data sets [3]. A comparative study between it and the hierarchical clustering algorithm was also conducted by Divya and Maniraj (2025) to highlight the strengths and weaknesses of each. It showed that K-means is more efficient with well-defined data clusters, while hierarchical clustering provides a clearer representation of hierarchical relationships between data [4]. In addition, Sinaga and Yang (2020) introduced the Unsupervised K-Means (U-K-means) algorithm, which improves on the traditional version by automatically determining the optimal number of clusters and reducing the impact of the initialization problem [5]. In the financial domain, K-Means was applied by Huang, Zheng, Li, and Che (2024), where it was used to cluster banking transaction data and identify abnormal patterns that may indicate fraudulent activities [6]. It has also been applied in the social field to assess social assistance eligibility and classify families by poverty levels and needs [7] . In the field of smart education, K-Means was employed by Lahmadi, El Khattabi, Rahhali, and Oughdir (2024) to classify students' learning styles and allocate educational resources according to their individual needs, thereby enhancing the effectiveness of adaptive educational systems [8]. Alzahrani, Meccawy, Samra, and El-Sabagh (2025) also addressed the application of K-Means in the field of e-learning, using student data from learning management systems (LMS) to discover weekly participation patterns. The results were validated using internal and external metrics, and the algorithm proved its ability to predict participation levels (low, medium, high) and link them to academic performance, making it an effective tool for supporting adaptive learning [9]. The dataset used in the analysis was obtained from Kaggle, a popular platform for sharing datasets and data science competitions [10]. Based on this background, this paper aims to review and apply the K-means algorithm to an unclassified dataset. We applied evaluation metrics to the algorithm to assess the performance of the clustering, identify optimal centers, and visually represent the results, thereby contributing to the understanding of the internal structure of the data and the extraction of hidden patterns.

The concept of the clustering:

Clustering is a method of unsupervised learning, where we deal with unlabelled data and try to discover patterns or internal structure in it. The main goal of clustering is to divide data into groups so that the elements within each group are as similar as possible, while being different from the elements in other groups. For example, if we have data on a group of clients relating to their ages, incomes, and number of purchases, clustering algorithms can help us discover the existence of client categories such as young people with average spending, or older people with high spending, and so on, without these categories being predefined.

Types of clustering algorithms:

1.k-Means algorithm is the best unsupervised clustering algorithm due to its simplicity and speed. It relies on determining the number of clusters (k) in advance. The idea behind the K-Means algorithm is very simple: the sample set is divided into K clusters according to the distance between the samples. Make the points in the clusters as close to each other as possible, and make the distance between the clusters as large as possible. In other words, it is an iterative algorithm that divides an unlabelled data set into different clusters so that each data set belongs to only one cluster with similar characteristics.

2.Hierarchical clustering is one of the clustering algorithms that does not require knowing the number of clusters in advance. The idea here is to build a hierarchical tree that shows how data can be gradually merged or divided. Clustering can start from the bottom, where each point is considered an independent cluster and then gradually merged, or from the top, where all points are considered a single cluster and then divided into smaller clusters. This type is suitable when we want to understand the relationships between clusters hierarchically.

3.The DBSCAN algorithm is based on the idea of density. It searches for dense areas of data to form clusters, and considers sparse or isolated points as noise. The advantage of this method is that it does not require prior knowledge of the number of clusters and is able to handle complex cluster shapes and easily detect outliers.

4.There is also the Mean-Shift algorithm, which gradually moves the centres of the clusters toward areas of higher density until they stabilize. This method is capable of automatically detecting the number of clusters, but it is usually slower than K-Means.

MATERIALS AND METHODS:

We have several auxiliary methods here:

In this study, the k-means algorithm was created using:

1.The Python programming language on the Spyder development environment, which is a powerful environment for scientific experiments and model creation.

2.Several software libraries (numpy, pandas, matplotlib.pyplot, scipy.io) were used to implement and program the algorithm.

3.A two-dimensional dataset consisting of 300 points (300,2).

4.Measures to evaluate the quality of the algorithm's results, namely cohesion and separation measures, in addition to the target function.

The data was taken from the Kaggle platform. The algorithm was applied as follows.

Steps of the k-means algorithm :

We applied the k-means algorithm to a dataset taken from the Kaggle platform, which is a two-dimensional dataset, i.e., (300,2). We loaded the data into the algorithm and manually selected the number of centers k, choosing K=3. The algorithm was applied in the following steps:

1.Center configuration: After manually selecting the number of groups k, we select random centers from the data sets, or what is known as the cluster center, which is the center of the group. However, at the beginning, the exact center of the data will be unknown, so we select random data points and identify them as centers for each group. The centers are configured using the following equation:

$$C^{(0)} = \left\{K^{(0)} C_1^{(0)}, C_2^{(0)}, \ldots\ldots, C\right\} R^n \ni C_j \qquad (1)$$

2.Cluster Assignment: After initialization, we assign each point to its nearest cluster center by calculating the Euclidean distance between the point and all cluster centers. The point is assigned to the cluster center with the smallest Euclidean distance among all centers, which is calculated using the following equation:

$$assign(xi) = arg_{j\{1,\ldots,k\}}\, min \left\| x_i - C_j^{(t)} \right\|^2 \qquad (2)$$

3.Centroid Update: After assigning each point to its nearest cluster center, we update the location of each cluster center. The new cluster center Cj is the arithmetic mean of all points within the cluster, calculated using the following equation:

$$C_j^{(t+1)} = \frac{1}{|S_J|} \sum_{\{x_i \in s_j\}} x_i \qquad (3)$$

4.Iteration: After updating the centers, the assignment process is repeated again and continues until there is stability or a predetermined number of iterations T. Each time, the points are assigned to the new centers, and then the new centers C^(t+1) are calculated based on the points assigned to them. At this point, the new centers become as close as possible to their true values, representing the shape of the clusters more accurately and representatively, and can be represented mathematically:

$$C^{(t+1)} \approx C^{(t)} \qquad (4)$$

5.Objective function: After each iteration, we calculate the objective function, where the algorithm seeks to minimize the sum of squares within clusters, or what is known as the Within-Cluster Sum of Squares (WCSS). This is the primary objective that the K-means algorithm seeks to minimize, through which we can determine whether the algorithm has improved or reached stability. It is calculated using the following equation:

$$j = \sum_{j=1}^{k} \sum_{x_i \in s_j} \left\| C_j - x_i \right\|^2 \qquad (5)$$

6.Algorithm evaluation: After implementing the algorithm, we added evaluation metrics to the algorithm, which evaluate the quality of the clusters:

a). Cohesion: This measure evaluates the quality of the cluster internally, where the distance between each point x within cluster j and the center c is calculated, and then the average of these distances is calculated. If the value is small, it means that the points are very close to the center, which means that the points are homogeneous. If the value is large, it means that the points are far apart and not homogeneous.

It is calculated using the following equation:

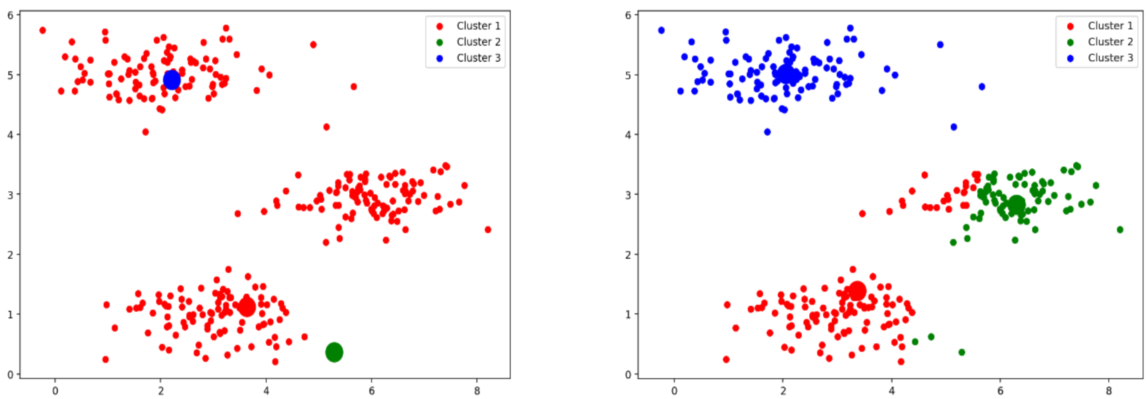
$$Cohesion(S_j) = \frac{1}{|S_J|} \sum_{\{x_i \in s_j\}} \left\| x_i - c_j \right\| \qquad (6)$$

b).Separation: This measure evaluates the value of separating clusters from each other by calculating the distance between the center Cj and other centers Cl, where the smallest distance (closest to it) is taken. If the value is large, it means that the clusters are separated and spaced apart, but if it is small, it means that the clusters are very close together, which leads to the possibility of overlap. It is calculated using the equation:

$$Separation(c_j) = min_{l \neq j} \left\| C_j - C_l \right\| \qquad (7)$$

Results and Discussion:

This study focused on applying the k-means algorithm using Python programming to a two dimensional dataset consisting of 300 points (300,2). where we manually set the number of clusters k=3. After applying the algorithm steps, we focused on tracking the evolution that occurs in the clustering process through iterations, where we experimented with the number of iterations T=6. The quality of the algorithm results was evaluated using the evaluation metrics (Cohesion)



and (Separation).

Figure 1: The top part of the diagram shows the initial initialization of the cluster centers as well as the distribution of points in iteration 1, where the red points represent the data, while the larger points (blue, red, and green) represent the cluster centers. It can be seen that all data points are grouped into a single cluster (red) with the red center. The bottom part shows that in iteration 2, the algorithm started to distribute the points among the clusters, where the data points appear in three colors. However, the algorithm did not cluster the points accurately, as overlaps between them can still be observed.

In addition to the objective function, which is calculated in each iteration. At the beginning of the first iteration, when the centers were randomly initialized, all points were clustered into one cluster, while the other two clusters did not contain any points (300,0,0), as shown in Figure 1. This was reflected in the high value of the objective function (J=4171.49) and also in the weakness of the Cohesion and Separation indices. However, in the second iteration, the algorithm began to distribute the points among the three clusters (154, 104, 42), as shown in Figure 2. The value of the objective function also decreased (J=1071.19), indicating that the algorithm began to improve the quality of the clustering by reducing the distance within the clusters. It was observed that the center moved towards new areas with higher point density, but there was still some overlap between the clusters.

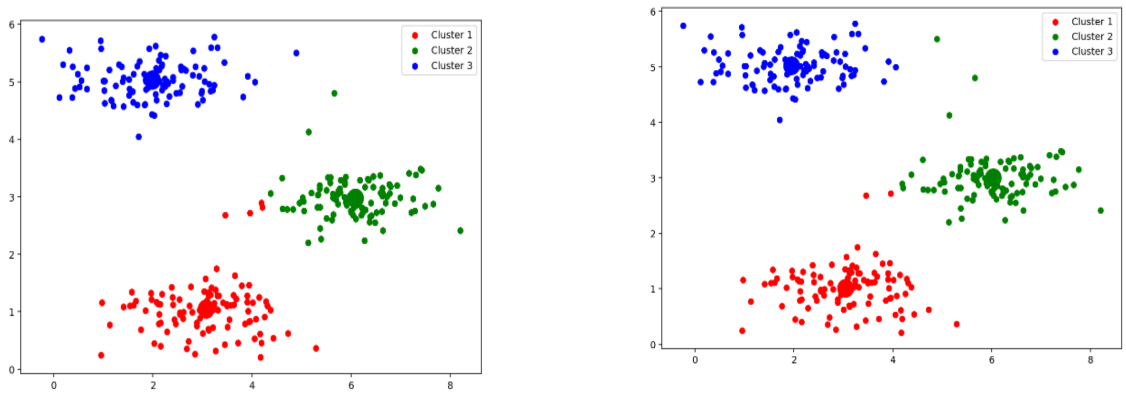


Figure 2: The top part of the diagram shows that in iteration 3, the distribution of points became more balanced compared to iteration 2 (77, 102, 121). In addition, the objective function decreased to (J = 566.99), and there was an improvement in the evaluation metrics, where the Cohesion measure decreased while the Separation measure increased, indicating a clear enhancement in the quality of the algorithm's results for achieving a more stable clustering. The bottom part of the diagram shows that in iteration 4, the points were distributed among the clusters (101, 100, 99), and the clusters became separated from each other, with each point grouped to its nearest cluster center. It was also observed that the objective function decreased further to (J = 268.312), along with an improvement in the evaluation metrics.

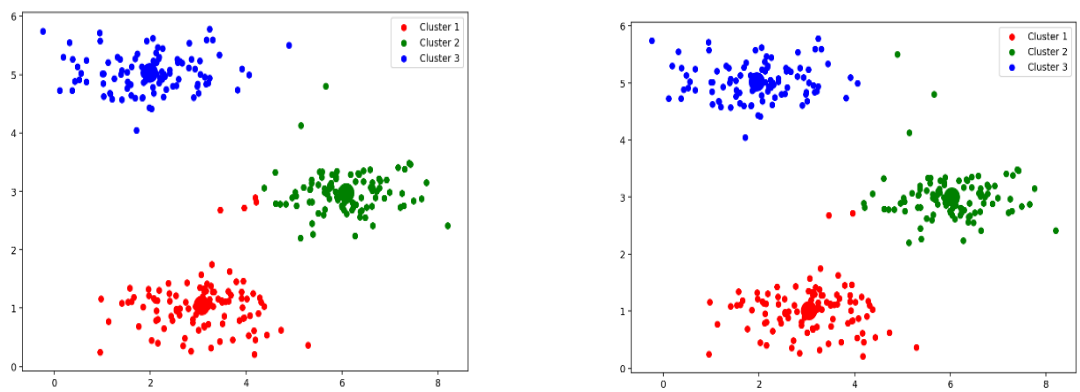


Figure 3: The top part of the diagram shows that in iteration 5, the points stabilized into clusters (102, 100, 98) with a slight change in the objective function (J = 266.65) and in the algorithm's evaluation metrics. The bottom part of the diagram shows that in iteration 6, the algorithm reached a stable stage, meaning no change occurred compared to iteration 5. The objective function remained at (J = 266.65), and the evaluation metrics (Cohesion and Separation) recorded their best results, indicating that the algorithm successfully clustered the data into three clusters.

The results of applying the algorithm to the dataset showed that it was able to cluster the data into K=3 clusters and achieved good performance after evaluating each cluster using the Cohesion measure, which indicates a decrease in the distances between points within each cluster. The algorithm also achieved good results when using the Separation measure, which reflects an increase in the distances between the three clusters, as illustrated in Figure 4. In addition, a significant reduction in the objective function was observed, as its value dropped from 4719.65 to 266.66 , as shown in Tables (1, 2, and 3). Each table presents the results obtained for each cluster.

Figure 4 also demonstrates the clear change in the Cohesion measure, where the values were initially very high (indicating weak clustering), then gradually decreased until stabilizing at values below 1. This reflects that the points are internally connected and well-integrated within each cluster, as shown in Figure 4 and Tables (1, 2, and 3). Moreover, Figure 4 highlights the noticeable change in the Separation measure, which was initially low but increased with each iteration. It should be noted that the Separation measure represents the distance between a cluster center and its nearest center, i.e., the average distance to the other centers in each iteration. The average Separation (avg Separation) was calculated for each iteration, as presented in Table 4 and illustrated in Figure 4.

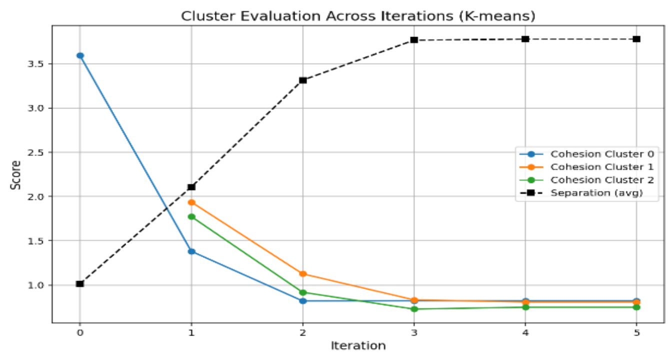


Figure 4: The graph shows the algorithm evaluation metrics over six iterations. The blue, orange, and green lines represent the cohesion of each cluster, which started high and then gradually decreased until it stabilized, indicating that the points became closer to the centers of their clusters. The black line shows Separation (avg), or how far apart the three clusters are from each other. We note that it was low at the beginning and began to rise with each iteration until it reached a final stable state, indicating that the clusters became more distinct.

These results show that the K-means algorithm is simple and easy to use, but it is sensitive to configuration in terms of choosing the number of clusters. We manually selected the number of clusters K=3, which is one of the fundamental limitations of the K-means algorithm, especially in large and high-dimensional data, affecting the performance of the algorithm and the clustering process. Therefore, previous studies have addressed this problem using several methods, including the study by Gowsic et al. (2024), which highlighted this problem. It used a combination of the Elbow method and the K-means algorithm, which proved to be accurate and significantly improved the performance of the algorithm compared to the traditional version [3].

TABLE 1: SHOWS THE SIZE OF POINTS, THE COHESION MEASURE, THE SEPARATION MEASURE, AND THE OBJECTIVE_J VALUE FOR CLUSTER 1.

Itera..	Clust.0:Size	Clust.0:Cohesion	Clust.0:Sep..	Object_ J
1	300	3.593	1.396	4719.65
2	42	1.382	2.920	1071.19
3	77	0.821	3.757	566.99
4	101	0.823	4.127	268.32
5	102	0.823	4.155	266.66
6	102	0.823	4.155	266.66

The table shows the results of the K-Means algorithm for group number (1) during six iterations when applied to a dataset consisting of 300 points, with the aim of analyzing the evolution of clustering performance across implementation stages. From the values shown, it can be seen that the cluster size decreased significantly from 300 points in the first iteration to 42 points in the second iteration as a result of the redistribution of points between clusters, then stabilized at around 102 points starting from the fourth iteration, indicating that the algorithm reached the convergence stage. The centers no longer changed. The cohesion value also decreased from 3.593 to 0.823, reflecting an increase in the homogeneity of points within the cluster, while the separation value increased from 1.396 to 4.155, which is evidence of improved clarity of the boundaries between clusters. Similarly, the objective function (Objective_J) recorded a significant decrease from 4719.65 to 266.66, indicating a continuous improvement in the quality of the clustering and a reduction in the squared distance between the points and their centers.

TABLE 2: SHOWS THE SIZE OF POINTS, THE COHESION MEASURE, THE SEPARATION MEASURE, AND THE OBJECTIVE_J VALUE FOR CLUSTER 2.

Itera..	Clus..1:Size	Clus..1:Cohesion	Clus..1:Sep….	Object_J
1	0	none	0.822	4719.65
2	104	1.938	1.699	1071.19
3	102	1.127	3.091	566.99
4	100	0.833	3.583	268.32
5	100	0.809	3.588	266.66
6	100	0.809	3.588	266.66

The table shows the results of the K-Means algorithm for cluster No.(2) through six iterations on a dataset containing 300 points. Initially, the cluster did not contain any points (0 points in the first iteration), then it began to receive points, reaching 104 points in the second iteration, and stabilizing at 100 points starting from the fourth iteration, reflecting the stability of the data distribution within the cluster. The Cohesion value decreased from 1.938 to 0.809, Separation increased from 1.699 to 3.588, and Objective_J decreased from 1071.19 to 266.66, indicating an improvement in the quality of the grouping, the differentiation of the cluster from other clusters, and the stability of the algorithm.

TABLE 3: SHOWS THE SIZE OF POINTS, THE COHESION MEASURE, THE SEPARATION MEASURE, AND THE OBJECTIVE_J VALUE FOR CLUSTER 3.

Itera..	Clus..2:Size	Clus..2:Cohesion	Clus..2:Sep..	Object_J
1	0	none	0.822	4719.65
2	154	2.15	1.699	1071.19
3	121	1.55	3.091	566.99
4	99	0.84	3.583	268.32
5	98	0.82	3.588	266.66
6	98	0.82	3.588	266.66

The table shows the results of the K-Means algorithm for cluster No. (3) through six iterations, on a dataset containing 300 points. Initially, the cluster did not contain any points (0 points in the first iteration), then it began to receive points, reaching 154 points in the second iteration, before stabilizing at around 98 points starting from the fifth iteration, reflecting the stability of the data distribution within the cluster. The Cohesion value decreased from 2.15 to 0.82, Separation increased from 1.699 to 3.588, and Objective_J decreased from 1071.19 to 266.66, indicating an improvement in the quality of the grouping, the differentiation of the cluster from other clusters, and the stability of the algorithm.

TABLE 4: SHOWS THE EVOLUTION OF THE SEPARATION MEASURE FOR EACH CLUSTER OVER SIX ITERATIONS AND THE CALCULATION OF THE AVERAGE SEPARATION (AVG) FOR EACH ITERATION, AS ILLUSTRATED IN FIGURE 4.

Itera..	Clus..0:Sep..	Clus..1:Sep..	Clus..2:Sep..	Sep..(avg)
1	1.396	0.822	0.822	1.013
2	2.920	1.699	1.699	2.106
3	3.757	3.091	3.091	3.313

4	4.127	3.583	3.583	3.764
5	4.155	3.588	3.588	3.777
6	4.155	3.588	3.588	3.777

The table shows the evolution of the separation metric for each cluster over six iterations, as well as the average separation for each iteration. We observe a clear increase in separation for all clusters: cluster 0 values rose from 1.396 to 4.155, cluster 1 from 0.822 to 3.588, and cluster 2 from 0.822 to 3.588, indicating that the clusters became more separated and distinct with repetitions. The overall average separation shows an increase from 1.013 to 3.777, reflecting an overall improvement in the clarity of the boundaries between clusters and the stability of the clustering process after the fifth repetition.

Conclusion :

In this research paper, we applied the K-means algorithm to a two-dimensional dataset taken from the Kaggle platform with a size of (300,2) in order to cluster and analyze this data. After entering the data into the algorithm and specifying the number of clusters K=3, the points were clustered and distributed among them .The algorithm was implemented according to systematic steps that included: first, initializing the centers, then clustering the points around the nearest center of the three centers by calculating the Euclidean distance, then updating the cluster centers, and finally using evaluation metrics to assess the performance of the algorithm. The cohesion measure showed that the algorithm achieved good homogeneity within the clusters. The separation measure showed that the algorithm achieved a good level of separation. The objective J function index was calculated, which helped evaluate the gradual improvement of the algorithm. The objective J function index also helped evaluate the gradual improvement, as its value decreased from 4719.65 in the first iteration to 266.65 in the sixth iteration. This indicates the stability of the algorithm and its success in clustering points within clusters. This paper highlights the power of the K-means algorithm in simplifying and clustering data. The use of evaluation metrics enhanced the clarity and credibility of the results. Although choosing the number of clusters is one of its most important limitations, especially with large and high-dimensional data, studies have shown that combining it with the Elbow Method improves performance and automatically determines the optimal number instead of manual determination. Overall, experiments have shown that the K-means algorithm is effective in clustering and unsupervised learning. **The scientific contribution** of this paper lies in its practical and explanatory aspects. It is a practical and educational addition that simplifies the understanding of the algorithm. It also relies on analyzing performance development through iterations using multiple evaluation metrics (Cohesion, Separation, Objective Function) simultaneously to evaluate the quality of the grouping, which adds scientific clarity in monitoring the performance of the algorithm.

future work :

1. Trying out the algorithm on real-world data that's bigger.

2. Using methods to figure out the K number instead of picking it manually, like the Elbow Method.

3. Using more evaluation metrics for the algorithm.

4. Select another algorithm and compare it with the K-means algorithm.

5. Apply the algorithm to different fields, such as image analysis, text analysis, or medical data, to test its effectiveness.

References

[1] M. Suyala and S. Sharma, “A Review on Analysis of K-Means Clustering Machine Learning Algorithm Based on Unsupervised Learning,” J. Artif. Intell. Syst., vol. 6, pp. 85–95, Apr. 2024, doi: 10.33969/AIS.2024060106.

[2] S. Naeem, A. Ali, S. Anam, and M. M. Ahmed, “An Unsupervised Machine Learning Algorithms: Comprehensive Review,” Int. J. Comput. Digit. Syst., vol. 13, no. 1, pp. 125–138, Apr. 2023, doi: 10.12785/ijcds/130172.

[3] G. Kowsic, M. S., S. L. Sakthivel, A. Puviyarasu, and R. M. Farook, “Enhanced Unsupervised K-Means Clustering Algorithm,” ShodhKosh: J. Vis. Perform. Arts, vol. 5, no. 1, 6pp. 1141–1150, Jan. 2024, doi: 10.29121/shodhkosh.v5.i1.2024.2867.

[4] G. Divya and V. Maniraj, “Exploring Clustering Techniques: Hierarchical vs. K-Means in Unsupervised Learning,” Int. J. Sci. Res. Eng. Trends, vol. 11, no. 1, pp. 775–783, Jan.–Feb. 2025.

[5] K. P. Sinaga and M. S. Yang, “Unsupervised K-Means Clustering Algorithm,” IEEE Access, vol. 8, pp. 80716–80727, 2020, doi: 10.1109/ACCESS.2020.2988796.

[6] Z. Huang, H. Zheng, C. Li, and C. Che, “Application of Machine Learning-Based K-Mean Clustering for Financial Fraud Detection,” J. Finance & Data Sci., pp. 1–10, 2024.

[7] A. F. Yulia, R. Ratnasari, and P. Bintoro, “K-Means Clustering in Determining the Eligibility of Recipients of Assistance for the Poor: Case Study of Village Sukoharjo III,” J. Socio-Econ. Dev. Stud., pp. 1–12, 2024.

[8] Y. Lahmadi, M. Z. El Khattabi, M. Rahhali, and L. Oughdir, “Optimizing Adaptive Learning: Insights from K-Means Clustering in Intelligent Tutoring Systems,” J. Educ. Technol. Res., pp. 1–15, 2024.

[9] N. Alzahrani, M. Meccawy, H. Samra, and H. A. El-Sabagh, “Identifying Weekly Student Engagement Patterns in E-Learning via K-Means Clustering and Label-Based Validation,” Electronics, vol. 14, no. 3018, pp. 1–27, Jul. 2025, doi: 10.3390/electronics14153018.

[10] Kaggle platform – Data source, available: https://www.kaggle.com/datasets