Implementation of Clustering using K-Means in Python

Ahmad Farhan AlShammari
Department of Computer and Information Systems
College of Business Studies, PAAET
Kuwait

ABSTRACT

The goal of this research is to develop a clustering program using k-means method in Python. Clustering helps to divide data into clusters (or groups) based on their features. K-means is used to assign the data points to the cluster of the closest center. Euclidean distance is used to measure the distances between the data points and the centers. K-means is an iterative method that continues in processing to update the centers until the final clusters are obtained.

The basic steps of clustering using k-means are explained: preparing data, initializing centers, computing labels (computing distances, finding minimum distance, and assigning labels), computing clusters, computing error function, updating centers, and plotting clusters.

The developed program was tested on an experimental dataset. The program successfully performed the basic steps of clustering using k-means and provided the required results.

Keywords

Artificial Intelligence, Machine Learning, Clustering, K-Means, Euclidean Distance, Centers, Labels, Clusters, Error Function, Python, Programming.

1. INTRODUCTION

In recent years, machine learning has played a major role in the development of computer systems. Machine Learning (ML) is a branch of Artificial Intelligence (AI) which is focused on the study of computer algorithms to improve the performance and efficiency of computer programs. [1-13].

Clustering is one of the important applications in machine learning. It is sharing the common knowledge between the following fields: machine learning, programming, data science, mathematics, statistics, and numerical methods [14-20].

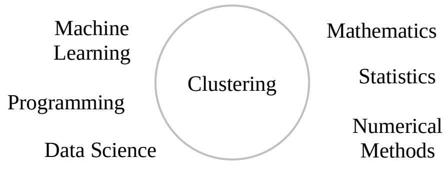


Fig 1: Field of Clustering

In this paper, clustering is applied using k-means method to divide data into clusters. K-means is used to measure and sort the distances between the data points and the centers, where the data points are assigned to the cluster of the closest center.

Clustering has a wide range of applications in different fields such as industry, business, education, marketing, advertising, medicine, public health, agriculture, environment, climate change, etc.

2. LITERATURE REVIEW

The related literature is reviewed to understand the major contributions in the field of clustering using k-means [21-25].

In general, algorithms in machine learning are divided into three main types: supervised, unsupervised, and reinforcement.

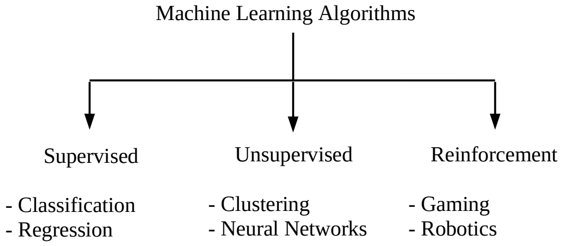


Fig 2: Types of Machine Learning Algorithms

Clustering is an unsupervised learning algorithm, where the data is not labeled and the algorithm is processed to divide data into groups based on their features.

K-means is a clustering method used to divide data into (*k*) clusters. It is performed by measuring and sorting the distances between the data points and the centers (or means). Then, the data points are assigned to the cluster of the closest center.

K-means was first developed in 1957 for signal processing by Stuart Lloyd at Bell Labs [26]. Then, it was published in 1965 by Edward Forgy [27]. It is sometimes called the Lloyd-Forgy method.

The fundamental concepts of clustering using k-means are explained in the following section.

Clustering:

Clustering is an important algorithm in machine learning. It helps to divide data into clusters (or groups) based on their features.

The concept of clustering is illustrated in the following diagram:

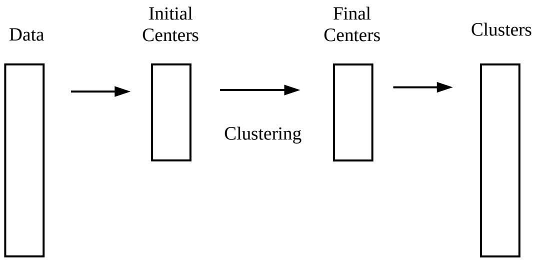


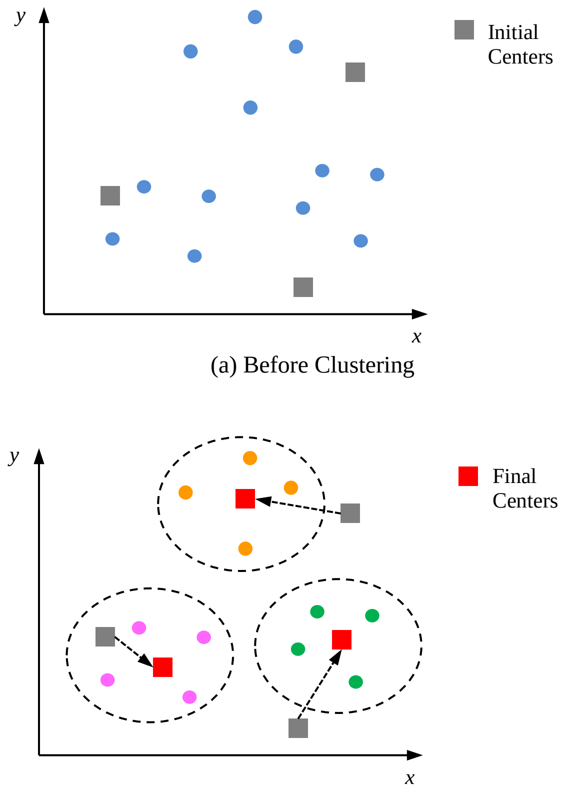
Fig 3: Explanation of Clustering

K-Means:

K-Means is a mathematical method used to divide data into (k) clusters (or groups) using the centers (or means). K-means is an iterative method that starts with giving initial values to the centers. The distances between the data points and the centers are measured using the Euclidean distance. Then, the distances are sorted in ascending order to find the minimum distance. After that, the data points are assigned to the cluster of the closest center.

The k-means method continues in processing to update the centers until the final clusters are obtained.

The concept of k-means is illustrated in the following diagram:



(a) Before Clustering

(b) After Clustering

Fig 4: Explanation of K-Means

Euclidean Distance:

Euclidean distance is the direct distance between two points. It is illustrated in the following diagram.

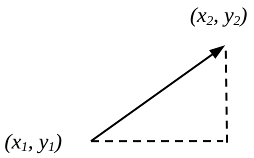


Fig 5: Explanation of Euclidean Distance

The Euclidean distance is computed by the following formula:

$$\text{Distance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \qquad (1)$$

The steps of clustering using k-means are explained in the following algorithm:

Algorithm 1: Clustering using K-Means Method

```
# Data
data = [ ... ]
# Number of Clusters
k = …
# Initial Centers
centers = [ ... ]
# Number of Iterations
nt = …
for i = 1 to nt do
    labels  = compute_labels(data, centers)
    clusters = compute_clusters(data, labels)
    error  = compute_error(clusters, centers)
    old_centers = centers
    centers  = update_centers(clusters)
    if (centers = old_centers) then
        exit
    end if
end for
```

Error Function:

Error function is used to evaluate the performance of the clustering model. It is defined as the average of the squared distances between the data points and the centers. It is computed by the following formula:

$$\text{Error} = \left(\frac{1}{n}\right)\sum_{i=1}^{k}\sum_{j=1}^{m}\left\|x_{ij} - c_i\right\|^2 \qquad (2)$$

Where: (x_{ij}) is the data point (j) in the cluster (i), (c_i) is the center of cluster (i), and (n) is the number of samples.

Clustering System:

The clustering system is explained in the following outline:

Input: Data (X, Y).
Output: Clusters.
Processing: The data is prepared for processing. At first, the centers are given initial values. Then, the distances between the data points and the centers are measured using the Euclidean distance. After that, the distances are sorted to find the minimum distance. At last, the data points are assigned to the cluster of the closest center and the final clusters are obtained.

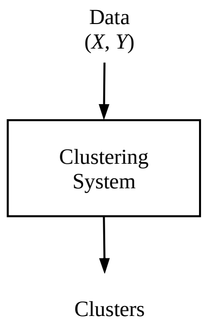


Fig 6: Diagram of Clustering System

Python:

Python [28] is a general high-level programming language. It is simple, easy to learn, and powerful. It is the most popular programming language for the development of machine learning applications.

Python provides many additional libraries for different purposes such as Numpy [29], Pandas [30], Matplotlib [31], NLTK [32], SciPy [33], and SK Learn [34].

In this research, the standard functions of Python are applied without using any additional library.

3. RESEARCH METHODOLOGY

The basic steps of clustering using k-means are: (1) preparing data, (2) initializing centers, (3) computing labels: (3.1) computing distances, (3.2) finding minimum distance, (3.3) assigning labels, (4) computing clusters, (5) computing error function, (6) updating centers, and (7) plotting clusters.

1. Preparing Data
2. Initializing Centers
3. Computing Labels:
    - 3.1. Computing Distances
    - 3.2. Finding Minimum Distance
    - 3.3. Assigning Labels
4. Computing Clusters
5. Computing Error Function
6. Updating Centers
7. Plotting Clusters

Fig 7: Steps of Clustering

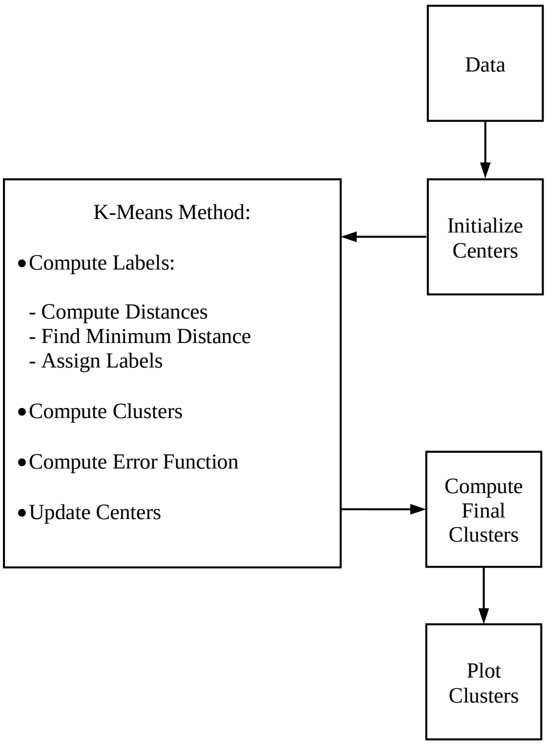


Fig 8: Flowchart of Clustering

The basic steps of clustering using k-means are explained in the following section.

1. Preparing Data:

The data is obtained from the original source and converted into list in the following form:

```
data = [[x0, y0],
        [x1, y1],
        [x2, y2],
        ...
        [xn-1, yn-1]]
```

2. Initializing Centers:

The centers are initialized to random values using the standard function random(). It is done by the following code:

```
def initialize_centers(data):
    centers = []
    for i in range(k):
        index = random.randint(0, n-1)
        centers.append(data[index])
    return centers
```

3. Computing Labels:

The labels are computed in three steps: computing distances, finding minimum distance, and assigning labels. It is done by the following code:

```
def compute_labels(data, centers):
    labels = []
    for i in range(len(data)):
        distances = compute_distances(data[i],
                    centers)
        min_distance = min(distances)
        label = distances.index(min_distance)
        labels.append(label)
    return labels
```

3.1. Computing Distances:

The distances are computed for all the data points with the centers using the Euclidean distance. It is done by the following code:

```
def compute_distances(point, centers):
    distances = []
    for i in range(len(centers)):
        distance = euclidean(point, centers[i])
        distances.append(distance)
    return distances
```

The Euclidean distance is computed using formula (1). It is done by the following code:

```
def euclidean(point1, point2):
    sum = 0
    for i in range(len(point1)):
        sum += (point1[i] - point2[i])**2
    return math.sqrt(sum)
```

3.2. Finding Minimum Distance:

The minimum distance is determined using the standard function min(). It is done by the following code:

```
min_distance = min(distances)
```

3.3. Assigning Labels:

The labels are assigned by the index of the minimum distance. It is done by the following code:

```
label = distances.index(min_distance)
labels.append(label)
```

4. Computing Clusters:

The clusters are computed according to the labels of the data points. It is done by the following code:

```
def compute_clusters(data, labels):
    clusters = []
    for i in range(k):
        clusters.append([])

    for i in range(len(data)):
        label = labels[i]
        clusters[label].append(data[i])
    return clusters
```

5. Computing Error Function:

The error function is computed for each iteration using formula (2). It is done by the following code:

```
def compute_error(clusters, centers):
    sum = 0
    for i in range(len(clusters)):
        for j in range(len(clusters[i])):
            sum += euclidean(clusters[i][j],
                   centers[i])**2
    return sum
```

6. Updating Centers:

The centers of the clusters are updated by the following code:

```
def update_centers(clusters):
    centers = []
    for i in range(len(clusters)):
        sumx = 0
        sumy = 0
        for j in range(len(clusters[i])):
            sumx += clusters[i][j][0]
            sumy += clusters[i][j][1]
        centers.append([sumx/len(clusters[i]),
                        sumy/len(clusters[i])])
    return centers
```

7. Plotting Clusters:

The final clusters are plotted using the matplotlib library. It is done by the following code:

```
plt.scatter(data_x, data_y)
for i in range(len(clusters)):
    cluster_x = transpose(clusters[i])[0]
    cluster_y = transpose(clusters[i])[1]
    plt.scatter(cluster_x, cluster_y)
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```

4. RESULTS AND DISCUSSION

The developed program was tested on an experimental dataset from Kaggle [35]. The program performed the basic steps of clustering using k-means and provided the required results. The program output is explained in the following section.

Data:

The data is printed as shown in the following view:

```
       X       Y
-----------------------
0:     15.0    39.0
1:     15.0    81.0
2:     16.0    6.0
3:     16.0    77.0
4:     17.0    40.0
5:     17.0    76.0
6:     18.0    6.0
7:     18.0    94.0
8:     19.0    3.0
9:     19.0    72.0
...
```

Initial Centers:

The initial centers are selected by random and printed as shown in the following view:

```
Initial Centers:
0:      [60.0, 50.0]
1:      [54.0, 46.0]
2:      [46.0, 46.0]
3:      [39.0, 36.0]
4:      [61.0, 42.0]
```

Labels:

The labels are computed and printed as shown in the following view:

```
Labels:
---------
0:      3
1:      2
2:      3
3:      2
4:      3
5:      2
6:      3
7:      2
8:      3
9:      2
...
```

Clusters:

The clusters are computed and printed as shown in the following view:

```
Clusters:
---------------------------
Cluster 0:
        0:      [69.0, 91.0]
        1:      [70.0, 77.0]
        2:      [71.0, 95.0]
        3:      [71.0, 75.0]
        ...
Cluster 1:
        0:      [39.0, 61.0]
        1:      [40.0, 55.0]
        2:      [40.0, 47.0]
        3:      [40.0, 42.0]
        ...
Cluster 2:
        0:      [15.0, 81.0]
        1:      [16.0, 77.0]
        2:      [17.0, 76.0]
        3:      [18.0, 94.0]
        ...
Cluster 3:
        0:      [15.0, 39.0]
        1:      [16.0, 6.0]
        2:      [17.0, 40.0]
        3:      [18.0, 6.0]
        ...
Cluster 4:
        0:      [70.0, 29.0]
        1:      [71.0, 11.0]
        2:      [71.0, 9.0]
        3:      [72.0, 34.0]
        ...
```

Final Centers:

The final centers are computed and printed as shown in the following view:

```
Final Centers :
0:      [86.538, 82.128]
1:      [55.088, 49.713]
2:      [25.727, 79.364]
3:      [26.304, 20.913]
4:      [87.750, 17.583]
```

Data and Final Clusters Plots:

The data (before clustering) is plotted as shown in the following chart:

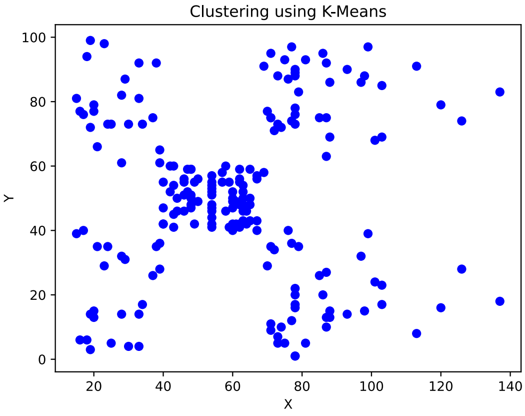


Fig 9: Data Plot

The final clusters are plotted as shown in the following chart:

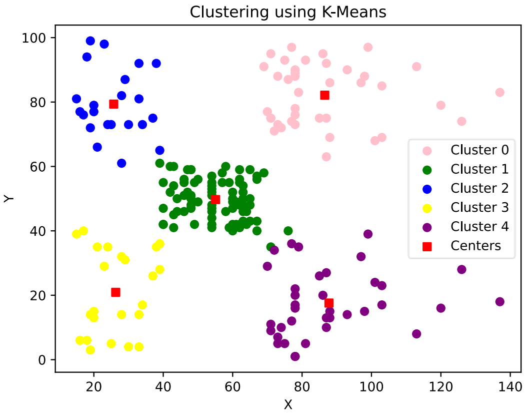


Fig 10: Final Clusters Plot

Error Function:

The error function is computed for each iteration and printed as shown in the following view:

```
t       Error
---------------------------
0       982.96
1       317.95145852256155
2       225.89996893097594
3       222.40388956108646
4       222.27238239839883
```

The error function is plotted as shown in the following chart:

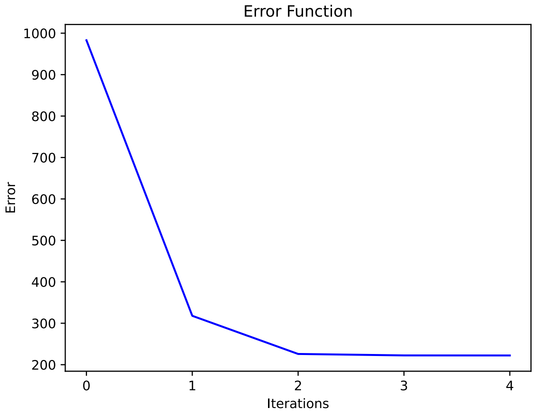


Fig 11: Error Function Plot

The plot shows that the error function is decreasing with iterations which indicates that the clustering model is converging to the optimal solution.

Number of Clusters:

The optimal number of clusters is determined using the "Elbow" Method which is (*k*=5). It is shown in the following chart:

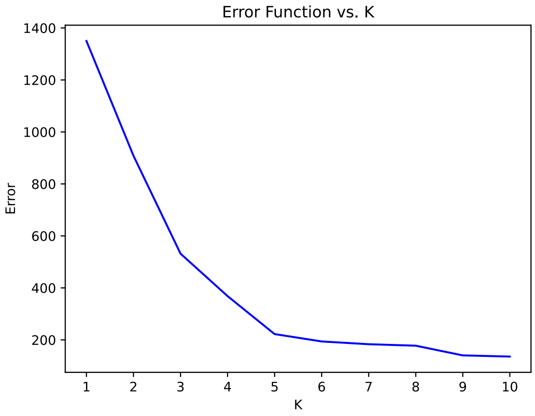


Fig 12: Error Function vs K Plot

In summary, the program output shows that the program has successfully performed the basic steps of clustering using k-means and provided the required results.

5. CONCLUSION

Machine learning is playing a major role in the development of computer systems. It helps to improve the performance and efficiency of computer programs.

Clustering is one of the important applications in machine learning. It helps to divide data into clusters (or groups). K-means is used to measure the distances between the data points and the centers using the Euclidean distance. Then, the distances are sorted to find the minimum distance. After that, the data points are assigned to the cluster of the closest center.

In this research, the author developed a program to perform clustering using k-means in Python. The developed program performed the basic steps of clustering using k-means: preparing data, initializing centers, computing labels (computing distances, finding minimum distance, and assigning labels), computing clusters, computing error function, updating centers, and plotting clusters.

The program was tested on an experimental dataset and provided the required results: centers, labels, clusters, and error function.

In future work, more research is needed to improve and develop the current methods of clustering using k-means. In addition, they should be more investigated on different fields, domains, and datasets.

6. REFERENCES

[1] Sammut, C., & Webb, G. I. (2011). "Encyclopedia of Machine Learning". Springer Science & Business Media.

[2] Jung, A. (2022). "Machine Learning: The Basics". Singapore: Springer.

[3] Kubat, M. (2021). "An Introduction to Machine Learning". Cham, Switzerland: Springer.

[4] Li, H. (2023). "Machine Learning Methods". Springer Nature.

[5] Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2016). "Machine Learning: Algorithms and Applications". Crc Press.

[6] Dey, A. (2016). "Machine Learning Algorithms: A Review". International Journal of Computer Science and Information Technologies, 7 (3), 1174-1179.

[7] Bonaccorso, G. (2018). "Machine Learning Algorithms: Popular Algorithms for Data Science and Machine Learning". Packt Publishing.

[8] Jo, T. (2021). "Machine Learning Foundations: Supervised, Unsupervised, and Advanced Learning". Springer.

[9] Chopra, D., & Khurana, R. (2023). "Introduction to Machine Learning with Python". Bentham Science Publishers.

[10] Müller, A. C., & Guido, S. (2016). "Introduction to Machine Learning with Python: A Guide for Data Scientists". O'Reilly Media.

[11] Raschka, S. (2015). "Python Machine Learning". Packt Publishing.

[12] Forsyth, D. (2019). "Applied Machine Learning". Cham, Switzerland: Springer.

[13] Sarkar, D., Bali, R., & Sharma, T. (2018). "Practical Machine Learning with Python". Apress.

[14] Han, J., Kamber, M., Pei, J. (2011). "Data Mining: Concepts and Techniques". Morgan Kaufmann, Burlington.

[15] Hand, D., Smyth, P. (2001). "Principles of Data Mining". MIT Press, Cambridge

[16] Kong, Q., Siauw, T., & Bayen, A. (2020). "Python Programming and Numerical Methods: A Guide for Engineers and Scientists". Academic Press.

[17] Unpingco, J. (2022). "Python for Probability, Statistics, and Machine Learning". Cham, Switzerland: Springer.

[18] Brandt, S. (2014). "Data Analysis: Statistical and Computational Methods for Scientists and Engineers". Springer.

[19] VanderPlas, J. (2017). "Python Data Science Handbook: Essential Tools for Working with Data". O'Reilly Media.

[20] James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). "An Introduction to Statistical Learning: With Applications in Python". Springer Nature.

[21] Oyewole, G. J., & Thopil, G. A. (2023). "Data Clustering: Application and Trends". Artificial Intelligence Review, 56(7), 6439-6475.

[22] Hartigan, J. A., & Wong, M. A. (1979). "A K-Means Clustering Algorithm". Applied Statistics, 28(1), 100-108.

[23] Wilkin, G. A., Huang, X. (2007). "K-Means Clustering Algorithms: Implementation and Comparison". In: Proceedings of the Second International Multi-Symposiums on Computer and Computational Sciences, pp. 133–136. IEEE.

[24] Pham, D. T., Dimov, S. S., & Nguyen, C. D. (2005). "Selection of K in K-means Clustering". Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 219(1), 103-119.

[25] Kodinariya, T. M., & Makwana, P. R. (2013). "Review on Determining Number of Cluster in K-Means Clustering". International Journal, 1(6), 90-95.

[26] Lloyd, S. P. (1957). "Least Squares Quantization in PCM". Bell Labs Paper. Published later in IEEE Transactions on Information Theory. (1957/1982), 18(11).

[27] Forgy, E. W. (1965). "Cluster Analysis of Multivariate Data: Efficiency versus Interpretability of Classifications". Biometrics. 21 (3): 768–769.

[28] Python: https://www.python.org

[29] Numpy: https://www.numpy.org

[30] Pandas: https:// pandas.pydata.org

[31] Matplotlib: https://www. matplotlib.org

[32] NLTK: https://www.nltk.org

[33] SciPy: https://scipy.org

[34] SK Learn: https://scikit-learn.org

[35] Kaggle: https://www.kaggle.com