Understanding K-Means Clustering: A Step-by-Step Guide

NAME – MUSKAN TEHRA

STUDENT ID - 23110806

1. Introduction

K- means is a machine learning algorithm that groups data into clusters based on similarity patterns. Unlike supervised learning, where models are trained with labeled data, K-Means is an unsupervised algorithm that identifies hidden patterns without predefined labels. It works by iteratively adjusting cluster centers (centroids) so that data points within a cluster remain close together while staying distinct from points in other clusters.

Dataset Overview

For this tutorial, we use the Iris dataset, a well-known dataset containing flower measurements from three different species. It consists of 150 samples with four numerical features:

- Sepal Length
- Sepal Width
- Petal Length
- Petal Width

Since the dataset naturally contains three species, K-Means clustering is an excellent choice to see how well the algorithm can separate the species without labels.

2. Understanding the K-Means Algorithm

The K-Means algorithm follows an iterative process to group data points into clusters. The steps are:

1. Select the number of clusters (K). The user defines the number of clusters to be formed.
2. Initialize K centroids. Centroids can be chosen randomly or through an optimized approach such as K-Means++.
3. Assign each data point to the nearest centroid. This is based on the Euclidean distance (or another distance metric).

4. Recalculate the centroids. The centroid of each cluster is updated by computing the mean of all data points assigned to that cluster.
5. Repeat steps 3 & 4 until centroids no longer change significantly (convergence).

Choosing the Best K-Value

Selecting the right number of clusters (K) is crucial for accurate clustering.
Two common methods to determine K are:

1. Elbow Method: Evaluates how compact clusters are (inertia/WCSS) and finds the "elbow point" where increasing K no longer provides significant improvement.
2. Silhouette Score: Measures how similar a data point is to its assigned cluster compared to other clusters.

3. Implementing K-Means Clustering in Python

Below is the Python implementation of K-Means clustering using the Iris dataset.

Step 1: Import Required Libraries

- numpy and pandas handle data operations.
- matplotlib.pyplot is used for visualization.
- sklearn.cluster.KMeans helps apply the K-Means algorithm.
- sklearn.datasets.load_iris loads the Iris dataset.
- sklearn.preprocessing.StandardScaler normalizes the dataset.

Step 2: Load and Normalize Data

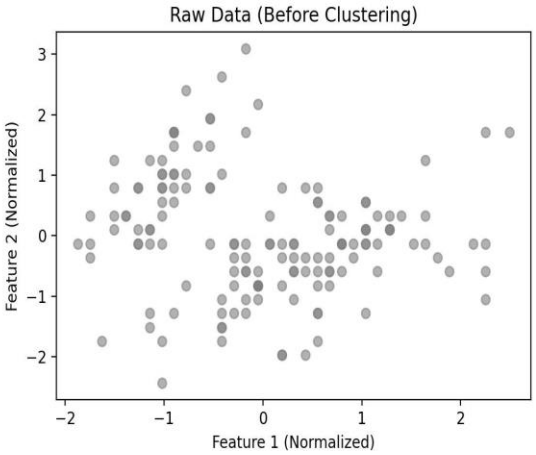
We load the Iris dataset and standardize the numerical values to ensure fair distance calculations.

normalizing the data because

- K-Means relies on distance calculations (Euclidean distance).
- Features with larger values might dominate clustering, making standardization necessary.

Step 3: Visualizing Raw Data (Before Clustering)

Before applying clustering, we visualize the raw data distribution.



Step 4: Applying K-Means Clustering

Now, we apply K-Means clustering with K=3 (since we know there are three species in the dataset).

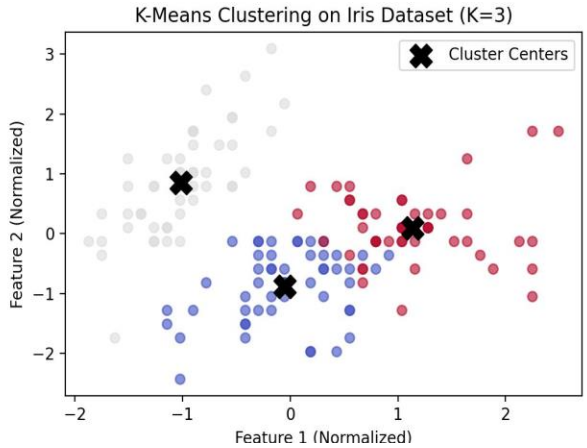
- n_clusters=3 sets the number of clusters.
- random_state=42 ensures reproducibility.
- n_init=10 runs K-Means 10 times to avoid poor initialization.

Step 5: Getting Cluster Labels & Centroids

- labels_ assigns a cluster to each data point.
- cluster_centers_ stores the centroid of each cluster.

Step 6: Visualizing the Clustered Data

After clustering, we visualize how the data points have been grouped.



Step 7: Finding the Optimal K Using the Elbow Method

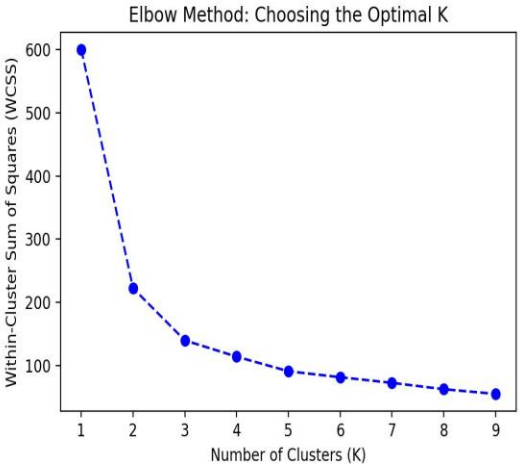
One way to determine the best value for K is the Elbow Method, which calculates the Within-Cluster Sum of Squares (WCSS) for different values of K.

The inertia_ attribute represents how compact the clusters are.

As K increases, WCSS decreases, but after a point, the reduction is insignificant.

Step 8: Plotting the Elbow Curve

We visualize how WCSS changes with different K values.



- The "elbow" in the curve represents the optimal K.
- In this case, the optimal number of clusters is K=3, which aligns with our dataset.

4. Conclusion

- K-means clustering is an effective unsupervised learning technique for finding patterns in data without predefined labels.
- The Elbow Method is a simple yet effective approach to determining the optimal number of clusters.
- The Iris dataset serves as a great example of how K-Means can group data points meaningfully.
- Understanding data preprocessing, feature scaling, and choosing K wisely are key factors in achieving good clustering performance.

5. References

Books & Research Papers

1. **MacQueen, J. (1967).** *Some methods for classification and analysis of multivariate observations.* Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability.
    a. This paper introduced the K-Means algorithm.
    b. Available at: https://projecteuclid.org/euclid.bsmsp/1200512992
2. **Lloyd, S. (1982).** *Least squares quantization in PCM.* IEEE Transactions on Information Theory, 28(2), 129-137.
    a. A foundational work on K-Means clustering.
3. **Hastie, T., Tibshirani, R., & Friedman, J. (2009).** *The Elements of Statistical Learning.* Springer.
    a. Chapter on clustering and K-Means.
    b. Free PDF: https://web.stanford.edu/~hastie/ElemStatLearn/

Online Courses & Tutorials

4. **Andrew Ng's Machine Learning Course – Coursera**
    a. Covers K-Means in a structured format.
    b. https://www.coursera.org/learn/machine-learning
5. **Scikit-learn Documentation: K-Means**
    a. Official implementation details of K-Means in Python.
    b. https://scikit-learn.org/stable/modules/clustering.html#k-means
6. **Wikipedia - K-Means Clustering**
    a. General introduction with explanations and references.
    b. https://en.wikipedia.org/wiki/K-means_clustering

Code & Implementation References

7. **K-Means Clustering in Python – Towards Data Science**
    a. A tutorial with explanations and code examples.
    b. https://towardsdatascience.com/k-means-clustering-explained-4528df86a120
8. **Machine Learning with Python – GeeksforGeeks**
    a. Step-by-step implementation in Python.
    b. https://www.geeksforgeeks.org/ml-k-means-algorithm/