UNIVERSITY OF DHAKA

Department of Computer Science and Engineering

CSE-4255 : Introduction to Data Mining and Warehousing Lab

Lab Report: Comparative Analysis of Clustering Algorithms (K-Means & DBSCAN)

Submitted By:

Name : Ahaj Mahhin Faiak

Roll No : 01

Submitted On:

JULY 7, 2025

Submitted To:

Dr. Chowdhury Farhan Ahmed

Md. Mahmudur Rahman

Contents

1 Introduction

Clustering is a fundamental task in unsupervised machine learning, where the goal is to group similar data points together without prior label information. Among the various clustering techniques, **K-Means** and **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) are two widely used algorithms with distinct approaches and assumptions.

K-Means is a *centroid-based algorithm* that partitions data into a predefined number of clusters by minimizing intra-cluster variance. It is simple, fast, and effective for convex and isotropic clusters but struggles with noise, outliers, and non-spherical shapes. On the other hand, DBSCAN is a *density-based algorithm* that groups together points that are closely packed, while marking points in low-density regions as noise. Unlike K-Means, DBSCAN does not require the number of clusters to be specified in advance and can discover clusters of arbitrary shapes, making it more robust to outliers and variations in density.

This report presents a comparative analysis of K-Means and DBSCAN across a diverse set of real-world datasets obtained from the UCI Machine Learning Repository. The comparison is based on two key clustering quality metrics: **Within-Cluster Sum of Squares (WCSS)** and the **Silhouette Coefficient**.

The Within-Cluster Sum of Squares (WCSS) measures the compactness of clusters and is defined as:

$$\text{WCSS} = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 \tag{1}$$

where k is the number of clusters, C_i is the set of points in cluster i, $\mathbf{x}$ is a data point, and $\boldsymbol{\mu}_i$ is the centroid of cluster i.

The Silhouette Coefficient evaluates the quality of clustering by measuring both intra-cluster cohesion and inter-cluster separation. For a single data point i, it is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{2}$$

where $a(i)$ is the average distance from point i to all other points in the same cluster, and $b(i)$ is the minimum average distance from point i to points in any other cluster. The overall Silhouette Coefficient is the mean of $s(i)$ over all data points.

By examining these metrics across datasets with varying characteristics, this report highlights the strengths and limitations of both algorithms, providing practical insights into their performance and suitability for different data distributions.

2 Implementation Details

This section outlines the implementation approach of the two clustering algorithms used in this study: **K-Means** and **DBSCAN**. Both algorithms were implemented from scratch using the Python programming language and applied to multiple datasets after necessary preprocessing.

2.1 K-Means Clustering

The K-Means algorithm is an iterative centroid-based method that aims to minimize the intra-cluster variance (WCSS). The pseudocode for K-Means is as follows:

Algorithm 1 K-Means Clustering

```
1: Initialize $k$ centroids randomly from the dataset.
2: repeat
3:     Assign each data point to the nearest centroid based on Euclidean distance.
4:     Recalculate the centroid of each cluster as the mean of assigned points.
5: until centroids do not change or maximum iterations reached
```

This process continues until the centroids stabilize, indicating convergence. The final output includes the cluster assignments and centroid locations. The Within-Cluster Sum of Squares (WCSS) and the Silhouette Coefficient are computed to evaluate the quality of clustering.

2.2 DBSCAN Clustering

DBSCAN is a density-based clustering algorithm that identifies clusters as high-density regions separated by low-density areas. Unlike K-Means, it does not require the number of clusters as input and is capable of detecting noise.

The pseudocode for DBSCAN is as follows:

Algorithm 2 DBSCAN Clustering

```
1: for each unvisited point $p$ in the dataset do
2:     Mark $p$ as visited.
3:     Find all points within distance $\varepsilon$ (neighborhood of $p$).
4:     if neighborhood size < MinPts then
5:         Mark $p$ as noise.
6:     else
7:         Create new cluster, expand it with density-reachable points.
8:     end if
9: end for
```

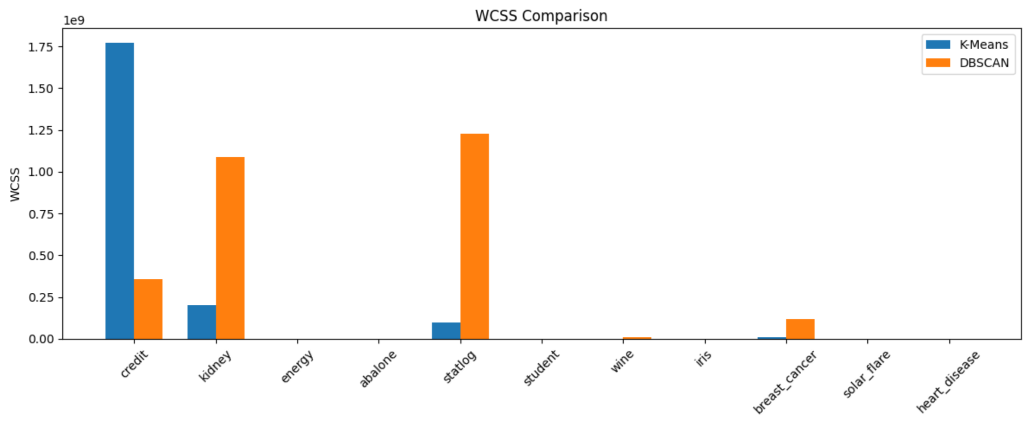
DBSCAN uses two parameters: ε (maximum neighborhood radius) and `MinPts` (minimum points to form a dense region). Points that do not belong to any cluster and are not density-reachable are labeled as noise. Similar to K-Means, the resulting clusters are evaluated using WCSS and the Silhouette Coefficient, ignoring noise points in the calculations.

2.3 Preprocessing and Parameter Estimation

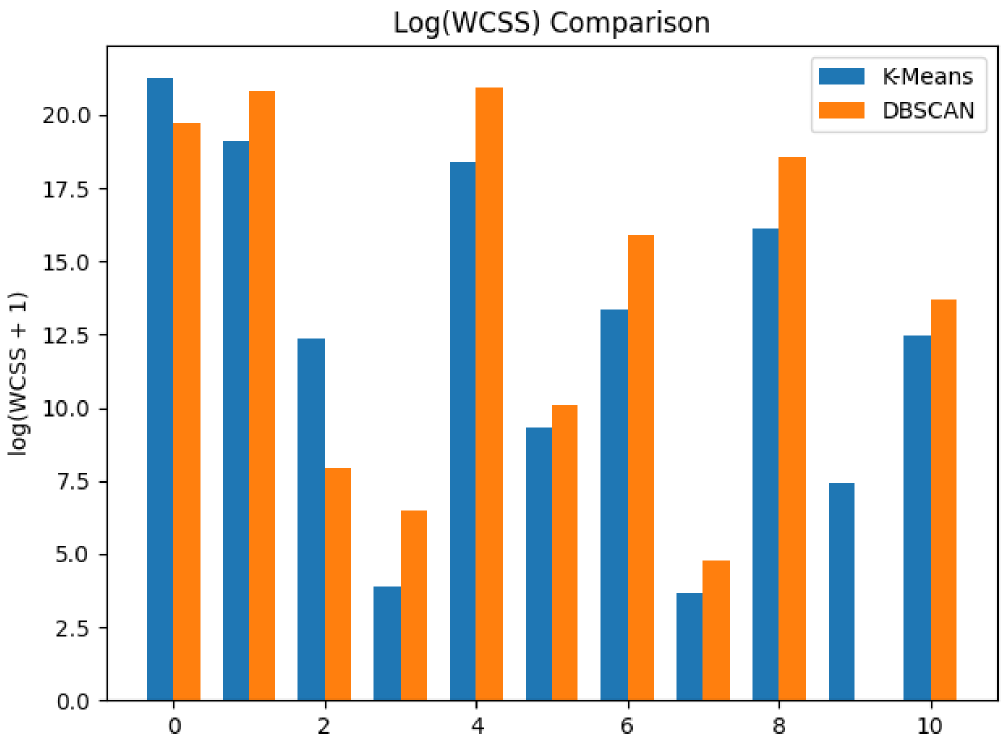
All datasets were preprocessed by encoding categorical attributes and filling missing values. Only numeric features were retained for clustering. For K-Means, the number of clusters k was estimated using a simple heuristic: $k \approx \sqrt{n}/2$, where n is the number of data points. For DBSCAN, parameters ε and `MinPts` were determined automatically using average k-distance from nearest neighbors.

3 Experimental Result

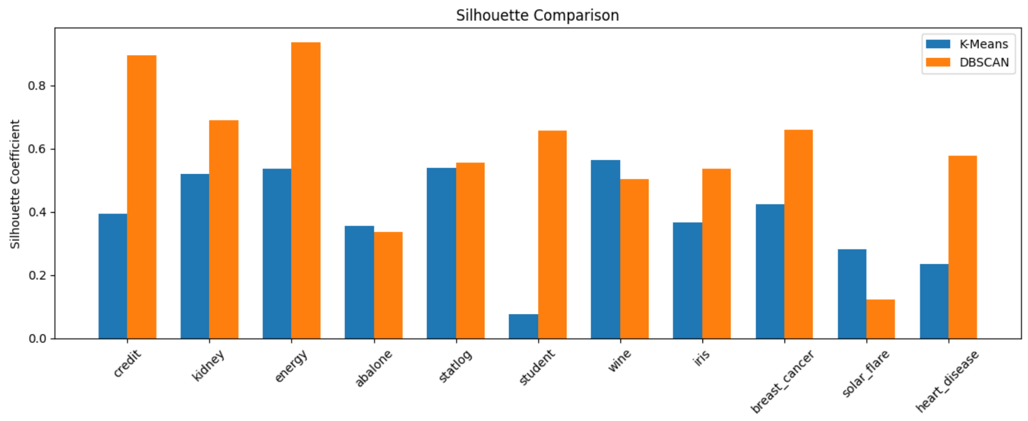
Here I present the experimental results of K-Means and DBSCAN:



(a) Within-Clustering Variation(WCSS) Comparison Graph



(a) log(WCSS) Comparison Graph



(a) Silhouette-Coefficient Comparison Graph

4 Conclusion

From the results, we observe the following trends:

- **DBSCAN often achieves higher Silhouette Coefficients** compared to K-Means, especially on datasets like `credit`, `energy`, `student`, and `breast_cancer`. This indicates better-defined and well-separated clusters due to DBSCAN's density-based nature.
- **K-Means performs better in terms of WCSS in several datasets**, particularly where clusters are spherical and balanced (e.g., `credit`, `abalone`, `wine`). However, DBSCAN shows strong WCSS performance in more complex datasets like `energy` and `student`.
- **DBSCAN is more robust to outliers and noise**, as reflected by improved Silhouette scores in noisy datasets, despite sometimes higher WCSS due to more dispersed cluster shapes.
- **K-Means is sensitive to the number of clusters k and initialization**, which affects convergence and final clustering quality. In contrast, DBSCAN's performance depends heavily on the choice of ε and `MinPts`, which we estimated automatically.

In summary, DBSCAN outperforms K-Means in terms of cluster quality on datasets with complex or non-globular shapes, while K-Means is more efficient and effective on well-structured datasets. The complementary strengths of both algorithms make them suitable for different types of data distributions.