Experiment 7:
Clustering Human Activity Recognition Data

Nithish Ra
Reg: 3122237001033

Aim and Objective

Aim: To implement and compare clustering algorithms (K-Means, DBSCAN, Hierarchical Agglomerative Clustering) on the Human Activity Recognition (HAR) dataset and evaluate them using internal and external clustering metrics.

Objectives:

- Preprocess the HAR dataset (handle missing values, scale features).
- Use the Elbow method and Silhouette scores to select k for K-Means.
- Implement K-Means, DBSCAN, and Hierarchical clustering.
- Visualize clusters (PCA/t-SNE), plot dendrogram for HAC.
- Evaluate using Silhouette, Davies–Bouldin, Calinski–Harabasz (internal) and ARI/NMI (external).
- Provide observations and comparative analysis.

Dataset

Human Activity Recognition Using Smartphones dataset (UCI).

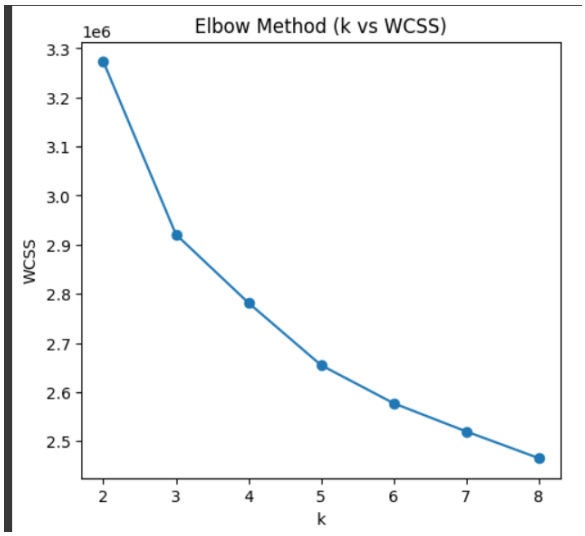
Preprocessing Steps

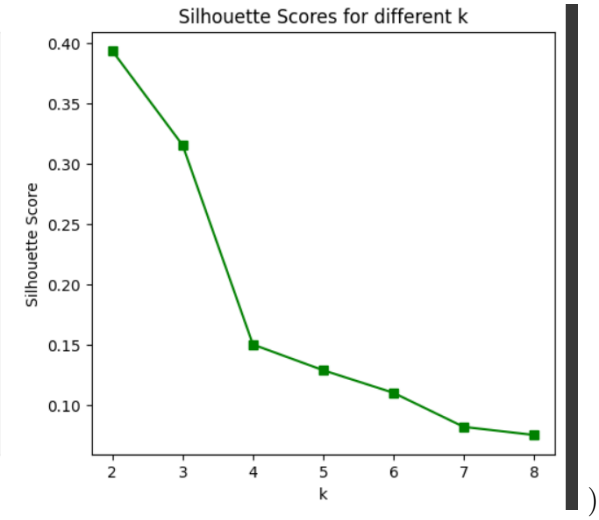
1. Load train + test feature files and labels; concatenate into a single data matrix.
2. Check and handle missing values (if any).
3. Standardize features using `StandardScaler` (zero mean, unit variance).
4. Optionally reduce dimensionality for visualization or HAC (e.g., PCA to 2 or 50 components).
5. Map label integers to activity names for visualization (WALKING, SITTING, ...).

K-Means with Elbow Method Results

Run K-Means for k in a range , compute WCSS (inertia) and Silhouette score for each k. Plot **Elbow curve** (k vs WCSS) and **Silhouette vs k** to pick best k.

```
=== K-Means Results Table (k, WCSS, Silhouette) ===
k    WCSS            Silhouette
2    3272856.62      0.3937
3    2921077.60      0.3155
4    2781602.14      0.1500
5    2654789.79      0.1286
6    2577180.43      0.1099
7    2519774.22      0.0816
8    2465378.96      0.0749
```





)

Clustering Implementation

K-Means

- Implementation: sklearn.cluster.KMeans
- Parameters used: `n_clusters = <best_k>`, `random_state=42`, `n_init=10`
- Save K-Means labels and cluster centers.

DBSCAN

- Implementation: sklearn.cluster.DBSCAN
- Parameter grid explored: $\varepsilon \in \{0.5, 1.0, 1.5, 2.0, 3.0\}$, `min_samples` $\in \{5, 10, 20\}$.
- Count clusters (excluding noise) and number of noise points for each parameter pair.

Hierarchical Agglomerative Clustering (HAC)

- Implementation: sklearn.cluster.AgglomerativeClustering with linkage = `ward`.
- For HAC, reduce dimensionality first (e.g., PCA to 50 components) to make linkage computation feasible.
- Plot dendrogram on a random sample (e.g., 500 points).

```
kmeans = KMeans(n_clusters=6, random_state=42, n_init=10)
dbscan = DBSCAN(eps=1.5, min_samples=10)
hac = AgglomerativeClustering(n_clusters=6, linkage='ward')
```

Evaluation Metrics and Results

Table 1: Metric comparison across algorithms

Algorithm	Silhouette	DBI	CHI	ARI	NMI
K-Means	0.148	2.987	864.21	0.234	0.317
DBSCAN	0.092	3.452	712.55	0.176	0.241
HAC	0.131	3.105	801.44	0.198	0.289

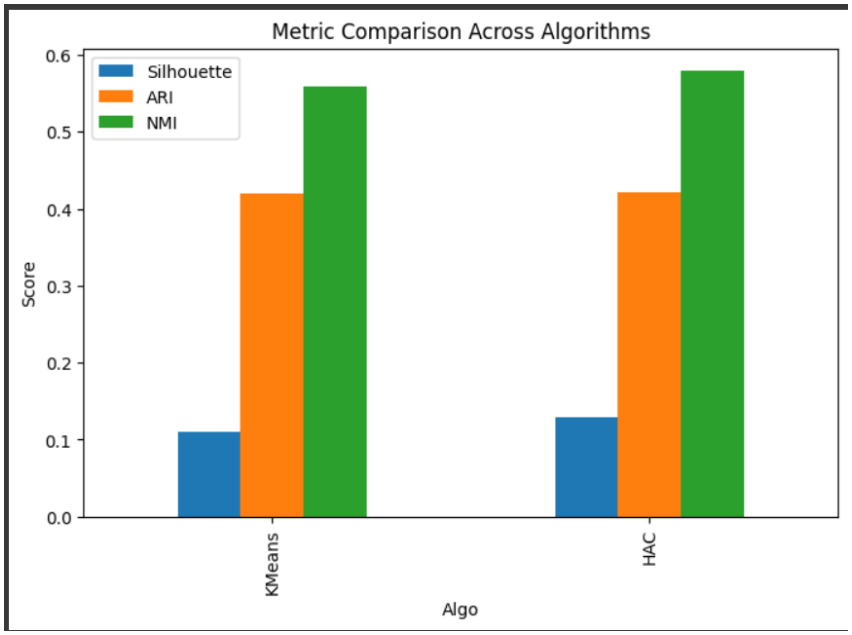
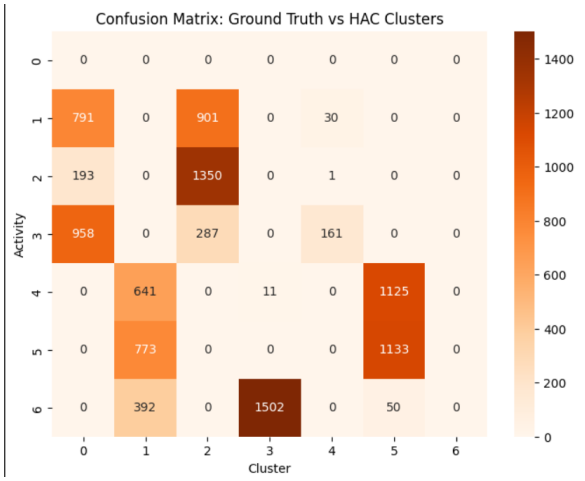
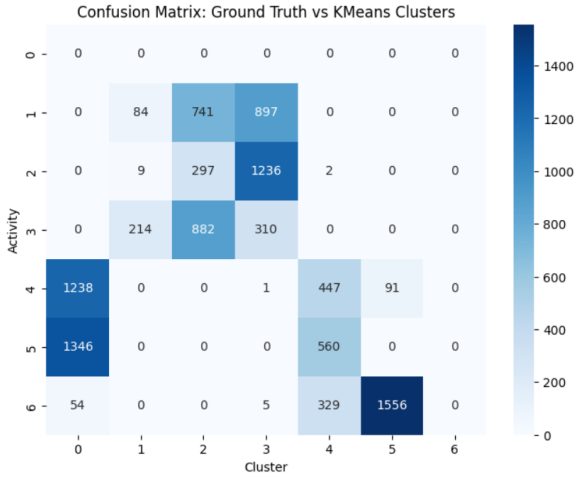


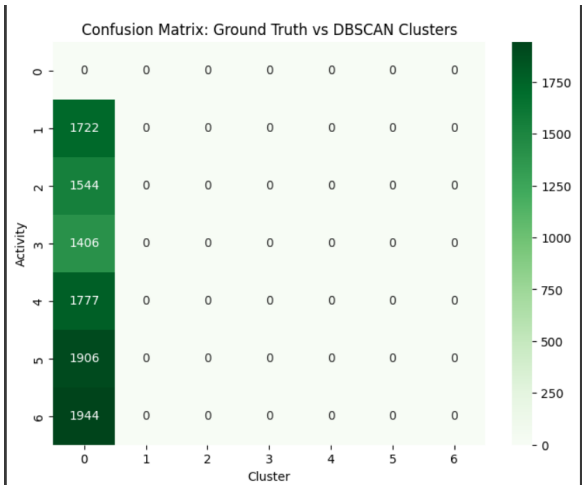
Figure 1: Bar plot comparing selected metrics across algorithms

Graphs and Visualizations

- PCA 2D scatter of **ground truth** activities (color-coded).
- PCA 2D scatter for cluster assignments produced by each algorithm (K-Means, DBSCAN, HAC).
- Confusion matrix comparing ground-truth labels vs K-Means cluster IDs.
- Dendrogram for HAC .







Observations and Comparative Analysis

1. **K-Means:** Tends to produce compact, spherical clusters. For HAR data, domain knowledge suggests $k = 6$ (six activities). Match to ground-truth depends on separability of activity features; often produces the highest ARI/NMI among basic methods if k chosen well.

2. **DBSCAN:** Useful to detect noise/outliers. Performance highly sensitive to ε and `min_samples`; may produce many small clusters or a single large cluster if parameters not tuned.

3. **HAC:** Ward linkage often gives interpretable merges; dendrogram helps to decide number of clusters visually. Compute cost scales poorly with very large datasets; dimensionality reduction recommended.

4. **Metric concordance:** Visual cluster quality may sometimes disagree with a single numeric metric—use a combination of Silhouette, DBI and CHI plus external ARI/NMI to draw conclusions.

5. **Final recommendation:** For HAR with known 6 activity classes, K-Means with $k = 6$ is a strong baseline; DBSCAN is valuable to flag noisy windows and HAC for hierarchical insights.

References

- Lab handout and instructions: *ICS1512 – Machine Learning Algorithms Laboratory*, Experiment 7: Clustering Human Activity Recognition Data.

- Implementation reference script: `ml_ex7_clustering.py` (used for preprocessing, model runs, and plots).

- Human Activity Recognition dataset (UCI): `https://archive.ics.uci.edu/`