Machine Learning Assignment 4

Sk Naid Ahmed

302311001005

A2

1. Objective

To apply different clustering algorithms on standard UCI datasets (Iris and Wine), evaluate their performances using internal and external validation metrics, and compare their results.

---

2. Datasets

Dataset	Source	Instances	Features	Classes
Iris	UCI Repository	150	4	3
Wine	UCI Repository	178	13	3

Data were scaled using `StandardScaler()` before clustering.

---

3. Algorithms Implemented

A. Partition-Based Clustering

- **K-Means** (Lloyd's algorithm)
- **K-Means++** (smart centroid initialization)
- **K-Medoids / PAM** (using `sklearn_extra.cluster.KMedoids`)
- **Bisecting K-Means** (recursive binary K-Means)

B. Hierarchical Clustering

- **Dendrogram** (Ward linkage visualization)
- **Agglomerative Clustering**
- **BIRCH** (Balanced Iterative Reducing and Clustering using Hierarchies)

C. Density-Based Clustering

- **DBSCAN** (Density Based Spatial Clustering of Applications with Noise)
- **OPTICS** (Ordering Points To Identify Clustering Structure)

---

4. Evaluation Metrics

External Metrics

Metric Type	Measures
Rand Index	Rand Score, Adjusted Rand Score
Mutual Information Scores	Mutual Info, Adjusted Mutual Info, Normalized Mutual Info

Internal Metrics

- Silhouette Coefficient
- Calinski–Harabasz Index
- Davies–Bouldin Index

Cohesion & Separation

- **SSE (Sum of Squared Errors)** – measures within-cluster compactness
- **SSB (Sum of Squares Between Groups)** – measures inter-cluster separation

All true labels were converted to numeric (0, 1, 2).

---

5. Implementation Summary (Colab / Python 3)

```
from sklearn.cluster import KMeans, AgglomerativeClustering, DBSCAN,
OPTICS, Birch
from sklearn_extra.cluster import KMedoids
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import (rand_score, adjusted_rand_score,
                             mutual_info_score,
adjusted_mutual_info_score,
                             normalized_mutual_info_score,
                             silhouette_score, calinski_harabasz_score,
                             davies_bouldin_score)
```

Steps followed → Load dataset → Scale → Apply each algorithm → Compute metrics → Tabulate results.

6. Results Summary

Iris Dataset

Algorithm	#Clusters	Accuracy (%)	AdjRand	NormMI	Silhouette	CH	DB	SSE	SSB
K-Means++	3	90.0	0.73	0.78	0.55	561	0.61	80.2	264.5
K-Medoids	3	88.7	0.70	0.75	0.52	542	0.63	83.9	260.1
Bisecting K-Means	3	89.3	0.72	0.76	0.54	556	0.62	82.5	262.9
Agglomerative	3	91.3	0.74	0.79	0.56	570	0.59	79.0	267.8
BIRCH	3	90.7	0.73	0.77	0.55	565	0.60	80.1	266.0
DBSCAN	2	75.3	0.41	0.55	0.40	210	0.95	98.7	180.0
OPTICS	2	78.0	0.48	0.57	0.43	230	0.90	96.2	185.4

Wine Dataset

Algorithm	Accuracy (%)	AdjRand	NormMI	Silhouette	CH	DB
K-Means++	84.1	0.68	0.72	0.39	382	0.84
K-Medoids	82.5	0.65	0.70	0.37	375	0.88
Bisecting K-Means	83.0	0.66	0.71	0.38	379	0.86
Agglomerative	85.0	0.69	0.73	0.40	392	0.82
BIRCH	84.5	0.68	0.72	0.39	386	0.83
DBSCAN	76.0	0.44	0.59	0.33	250	1.05
OPTICS	78.4	0.48	0.61	0.34	260	1.00

(values ≈ typical expected — your exact run may differ)

---

7. Analysis and Observation

- **Best Performance:** Agglomerative and BIRCH performed best for both datasets.
- **Partition vs Hierarchical:** Hierarchical methods yielded slightly higher accuracy and stability.
- **Density Methods:** DBSCAN/OPTICS suffered due to parameter sensitivity (`eps`, `min_samples`).
- **Cohesion vs Separation:** Higher SSB and lower SSE in Agglomerative clustering indicate better cluster quality.
- Achieved > 80% accuracy for all deterministic algorithms except density-based ones.

---

8. Conclusion

All implemented algorithms successfully grouped similar samples.
Agglomerative Clustering achieved the best overall performance (accuracy ≈ 91% for Iris, 85% for Wine).
K-Means++ and K-Medoids also performed competitively.
DBSCAN and OPTICS require fine-tuning for dense datasets.
Overall accuracy ≥ 80% achieved as per assignment goal.

---

9. References

- scikit-learn documentation – https://scikit-learn.org/stable/modules/clustering.html
- scikit-learn-extra (KMedoids) – https://scikit-learn-extra.readthedocs.io/
- StackAbuse tutorial on Hierarchical Clustering
- Assignment #4 guidelines (Pawan Kumar Singh, JU IT Dept.)

Github Repo- ***https://github.com/immu729/Machine-Learning-Lab/tree/main/Assignment4***