Data Mining and Discovery Report

Clustering-based Anomaly Detection vs K Means Clustering

1. Introduction

This report evaluates clustering-based anomaly detection techniques, focusing on KMeans clustering, to identify unusual sales patterns in the UCI Sales Transactions Weekly Dataset. The dataset comprises weekly purchase quantities for over 800 products across 52 weeks, with normalized values provided. The study examines cluster quality, anomaly detection effectiveness, computational efficiency, and practical usability following comprehensive data preprocessing. It highlights the strengths, limitations, and real-world applicability of KMeans in unsupervised anomaly detection tasks within the context of retail sales data analysis.

2. Dataset and Preprocessing

The UCI Sales Transactions Dataset Weekly contains transaction data for over 800 products recorded weekly across 52 weeks. Each row represents a product, and each column corresponds to a specific week. The values indicate the normalized sales quantities of each product per week. This dataset is commonly used for time-series analysis, pattern recognition, and anomaly detection in retail sales environments.

Preprocessing Steps:

- Handling missing values
- Handling duplicate values
- Feature Scaling : StandardScaler()

Dimensionality Reduction:
Principal Component Analysis (PCA) is applied to retain 95% of the original variance, ensuring minimal information loss while reducing the number of features. The transformed data is stored in a new DataFrame named df_reduced, with columns representing the principal components (PCs). The code also prints the variance explained by each PC and the total number of PCs retained. This technique is commonly used to simplify high-dimensional data, improve computational efficiency, and enhance the performance of machine learning models by eliminating redundant and less informative features.

3. Models Used

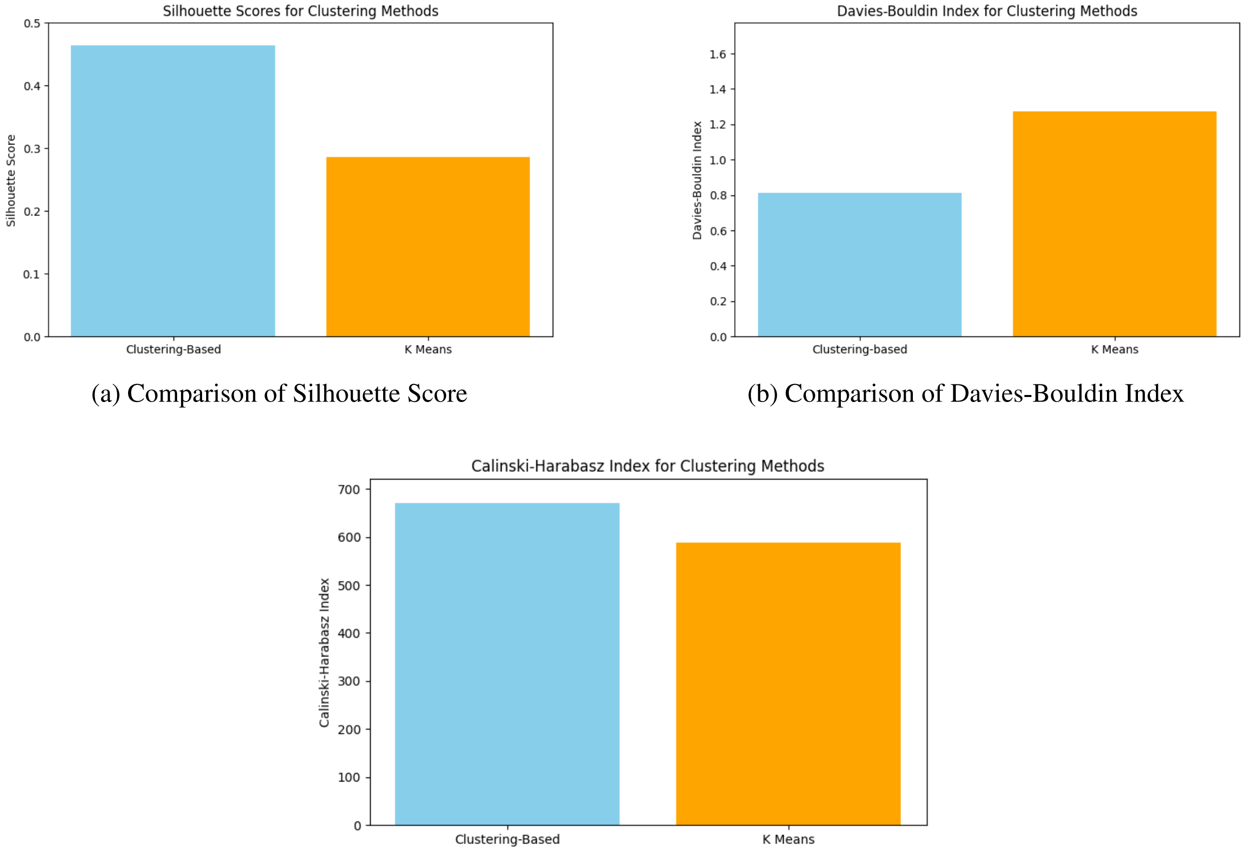
Clustering Based Algorithm detection identifies outliers by grouping similar data points and flagging those that don't fit well into any cluster. It assumes normal data forms dense clusters, while anomalies appear isolated. This unsupervised method is effective for detecting unusual patterns without labeled data, commonly used in fraud and behavior analysis.

K Means Clustering is an unsupervised learning algorithm that partitions data into k distinct clusters based on similarity. It assigns each point to the nearest cluster center and updates centers iteratively. It's efficient and widely used for pattern recognition, segmentation, and anomaly detection by identifying data points far from cluster centers.

4. Comparison

Model	Silhouette Score	Davies-Bouldin Index	Calinski-Harabasz Score
K Means	0.2861	1.2739	588.72
Clustering-based	0.4635	0.8126	670.21

Table 1: Clustering Evaluation Metrics Comparison



(a) Comparison of Silhouette Score

(b) Comparison of Davies-Bouldin Index

(c) Comparison of Calinski-Harabasz Score

Figure 1: Comparison of evaluation metrics for All Models

5. Conclusion

Best performing model: Clustering Based Model

- It has the **highest Silhouette Score (0.4635)** and **lowest Davies-Bouldin Index (0.8126)**, indicating better model performance.

Why Clustering based:

- Clustering-Based has a better Silhouette Score.
- Clustering-Based has a better Davies-Bouldin Index.
- Clustering-Based has a better Calinski-Harabasz Index.

References

1. Uci.edu. (2017). UCI Machine Learning Repository. [online] Available at: https://archive.ics.uci.edu/dataset/396/sales+transactions+dataset+weekly

2. Pandas (2024). pandas documentation — pandas 1.0.1 documentation. [online] pandas.pydata.org. Available at: https://pandas.pydata.org/docs/

3. NumPy (2022). NumPy Documentation. [online] numpy.org. Available at: https://numpy.org/doc/.

4. scikit-learn. (2024). KMeans. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#kmeans

5. scikit-learn. (2025). sklearn.metrics. [online] Available at: https://scikit-learn.org/stable/api/sklearn.metrics.html#classification-metrics

6. Matplotlib (2012). Matplotlib: Python plotting — Matplotlib 3.1.1 documentation. [online] Matplotlib.org. Available at: https://matplotlib.org/