Clustering Analysis using Human Activity Data

Umar Faruk Abdullahi[1]

Abstract

In this project, we build and evaluate unsupervised machine learning models using the UCI Human Activity Recognition dataset. The models are first evaluated without dimensionality reduction on the dataset and subsequently, newer iterations of the models are fitted after applying dimensionality reduction. The project intricately details the dataset, preprocessing, methods used to find optimal parameters for the models, impact of dimensionality reduction on the clustering process and a comparative study between the models.

Link to Google Colab.

Keywords: Data Science, Machine Learning, Unsupervised Learning, Clustering

1. Introduction

In recent years, the ubiquity of mobile and wearable devices equipped with an array of sensors has given rise to the generation of large amounts of data, opening new avenues for diverse applications spanning healthcare monitoring, fitness tracking, smart environment management, and beyond. Among the studies harnessing this wealth of data, one notable endeavor is the University of California Irvine (UCI) Human Activity Recognition project. This experiment involved the collection of data on physical activities through smartphones worn by 30 volunteer participants, yielding a rich dataset ripe for analysis.

By delving into this dataset and applying advanced clustering techniques, researchers can unravel hidden structures within human activity data, paving the way for understandings of behavior patterns, personalized recommendations, and innovations in healthcare, fitness, and beyond. This project aims to delve into the depths of the UCI Human Activity Recognition dataset, utilizing clustering analysis to extract meaningful insights. The project's objective is to use different clustering and dimensionality reduction techniques in a quest for optimization and better results.

2. Data

In this phase of the project, we identify the type of the data and its distribution. The dataset is a publicly available well known dataset for Human Activity Recognition as mentioned above: Uci HAR. The data collected contained numerous files containing the various parts of the dataset such as:

- Features
- Inertia Signals
- Activity Labels
- Train Data
- Test Data

The dataset was pre-processed and merged into a cumulative dataset to aid our project and its analysis. As this project is an unsupervised machine learning problem, the labels of the data were only used to provide information on the data distribution and evaluate the performance of the trained clustering algorithms.

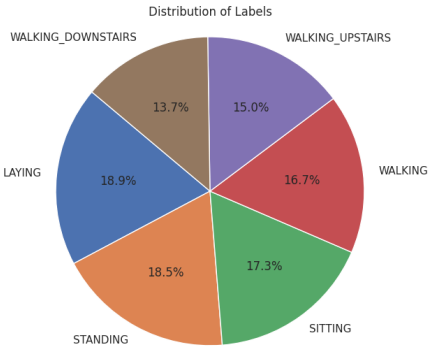

Figure 1: Activity distribution

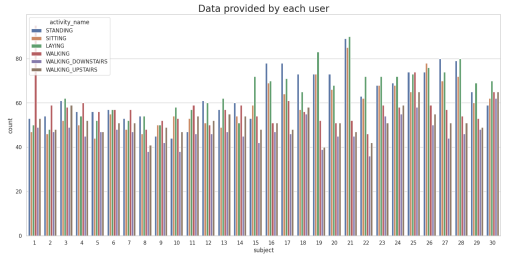

Figure 2: User representation

Algorithm 1 k-means algorithm

1: Specify the number k of clusters to assign.
2: Randomly initialize k centroids.
3: **repeat**
4: **expectation:** Assign each point to its closest centroid.
5: **maximization:** Compute the new centroid (mean) of each cluster.
6: **until** The centroid positions do not change.

Figure 3: k-Means

2.1. Data Processing and Visualization

After applying the pre-processing mentioned above, the data was inspected for missing values and statistical properties. The data contained no missing values and subsequently, scaling was applied on the data before it was used to train the clustering algorithms. Scaling before clustering is crucial for ensuring the accuracy and effectiveness of the clustering process. When dealing with datasets containing features with different scales and units, clustering algorithms may be heavily influenced by variables with larger magnitudes, leading to biased results and inaccurate cluster assignments.

Also, scaling improves the performance of distance-based clustering algorithms. Methods such as K-means or hierarchical clustering rely on distance metrics to determine the similarity between data points. It also enhances the convergence speed and stability of clustering algorithms. Algorithms like K-means converge faster and produce more stable cluster assignments when applied to scaled data.

The distribution of the datasets activities as well as representation of data from every participant was inspected to ensure a balanced dataset free from bias. As it is well know, biasness in the data will lead to inaccurate results and may affect all our subsequent processes.

3. Clustering Analysis

After inspecting the dataset and applying the preprocessing technique, the next step of the project was training a clustering model on the dataset. For this project, two unsupervised machine learning clustering models are used:

1. **KMeans Clustering**: KMeans Clustering: KMeans clustering is an unsupervised machine learning algorithm used for partitioning a dataset into a predetermined number of clusters. The algorithm works by iteratively assigning each data point to the nearest cluster centroid and then updating the centroids based on the mean of the data points assigned to each cluster. This process continues until the centroids no longer change significantly or a predefined number of iterations is reached.
2. **DBSCAN**: DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is also a clustering algorithm in machine learning designed to identify clusters of varying shapes and sizes in a dataset containing noise and outliers. Unlike traditional clustering algorithms like K-means, DBSCAN does not require specifying the number of clusters in advance. Instead, it groups together data points that are closely packed, forming dense regions separated by regions of lower density.

In every machine learning project, just training the model isn't sufficient. Informative and qualitative metrics must also be taken into account to evaluate the performance of the model. For this project, the following

Algorithm DBSCAN(D, $dist$, ϵ, MinPts)

Input:
D: a set of data points $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$
$dist(\mathbf{x}, \mathbf{x}')$: a distance function
ϵ: radius of a neighborhood
MinPts: the minimum number of points required to form a dense region

Initialize the labels of all the data points to be Unvisited
$C \leftarrow 0$ ▷ The current cluster id
for each data point $\mathbf{x} \in D$ **do**
  if label($\mathbf{x}$) $\neq$ Unvisited **then**
    continue
  label($\mathbf{x}$) $\leftarrow$ Visited
  $N \leftarrow$ RegionQuery(D, $dist$, $\mathbf{x}$, ϵ) ▷ Find neighbors of $\mathbf{x}$
  if $|N| <$ MinPts **then** ▷ Check if $\mathbf{x}$ is a core point
    label($\mathbf{x}$) $\leftarrow$ Noise
  else
    $C \leftarrow C + 1$ ▷ Form a new cluster
    ExpandCluster(D, $dist$, $\mathbf{x}$, N, C, ϵ, MinPts)

Figure 4: DBSCAN

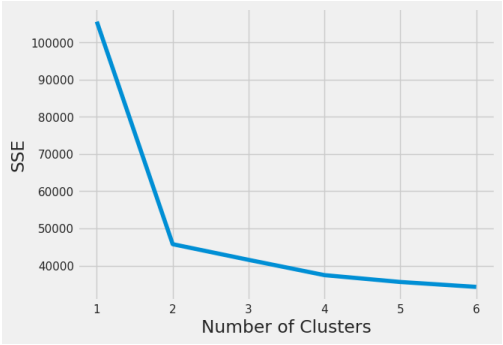

Figure 5: KMeans Cluster scores - Elbow method

metrics for evaluating clustering analysis were employed:

1. **Silhoutte Score**: The silhouette score is a metric used to evaluate the quality of clustering in unsupervised learning. It quantifies how similar an object is to its own cluster compared to other clusters. The silhouette score ranges from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. A score close to 1 suggests dense, well-separated clusters, while a score near 0 indicates overlapping clusters. Conversely, a negative silhouette score implies that the object may have been assigned to the wrong cluster.
2. **Davies Bouldin Score**: The Davies-Bouldin Index (DBI) is a clustering evaluation metric that quantifies the compactness and separation of clusters in a dataset. It measures the average similarity between each cluster's centroid and the centroids of its nearest neighboring clusters, normalized by the sum of the intra-cluster distances. A lower DBI value indicates better clustering, with tight, well-separated clusters. The DBI offers a straightforward interpretation: the closer the value is to zero, the better the clustering performance.

3.1. No Dimensionality Reduction

In the first step of the process, the clustering algorithms are applied on the dataset wihout using a dimensionality reduction technique. This is to understand the nature of the base clustering algorithms and find the optimal parameters for the methods.

In this stage, we use a RandomForestClassifier on the dataset (since it has ground truth), to understand the most important features in the dataset before visualizing our clustering. Due to the high dimensionality of the data, the selected features will aid in visualizing the data in a simple manner before applying dimensionality reduction techniques.

This step is only possible because the dataset already contains ground truth labels. In most scenarios of unsupervised learning, the dataset does not contain labels and this technique cannot be employed.

1. KMeans: In the KMeans model, the first step was to identify the optimal number of clusters. The elbow method was employed to find the cluster size with the lowest level of change. The result is shown in Figure 5.
   After choosing the number of clusters - **2**, which is the only parameter for the KMeans algorithm. The model was fitted and the results were visualized based on the features selected by our model above. However, interpreting the visualization was challenging due to only having two features from the hundreds of the data. This further underscores the need to use a dimensionality reduction technique in order to get a better visualization.
   The PCA technique was chosen. Principal Component Analysis (PCA) is a dimensionality reduction technique widely used in data analysis and machine learning. Its primary objective is to transform high-dimensional data into a lower-dimensional space while retaining as much of the original variability as

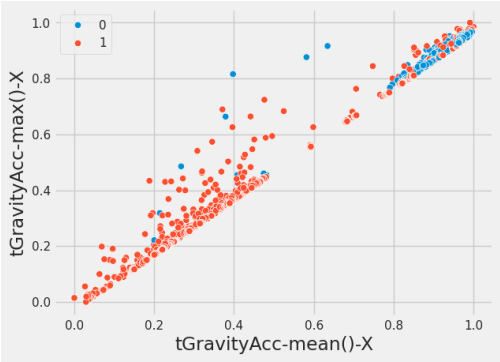

Figure 6: KMeans Visualization - No Dimensionality Reduction

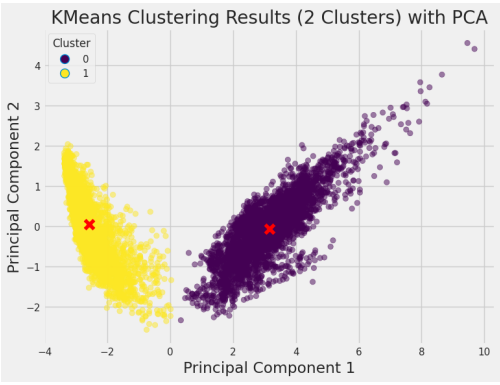

Figure 7: KMeans Visualization PCA

possible. PCA achieves this by identifying the directions, or principal components, along which the data varies the most. These principal components are orthogonal to each other, meaning they capture different aspects of the variability present in the data. By projecting the data onto a subset of these principal components, PCA effectively reduces the dimensionality of the dataset while preserving the most significant patterns and structures

The results are shown in Figure 7 and Figure 8:

2. DBSCAN: Unlike the KMeans clustering algorithm where the number of clusters is provided, DBSCAN finds the optimal number of clusters by itself. However, it uses two key hyperparameters to achieve this:

   (a) Epsilon: This is the most important hyperparameter of the DBSCAN algorithm. Epsilon defines the radius within which the algorithm searches for neighboring points around a core point. In other words, it determines the maximum distance between two points for them to be considered as part of the same neighborhood. Points within this radius are considered directly density-reachable from each other. Adjusting epsilon allows for the customization of the cluster density sensitivity: smaller values result in denser clusters, while larger values lead to more points being included in clusters, potentially merging neighboring clusters.

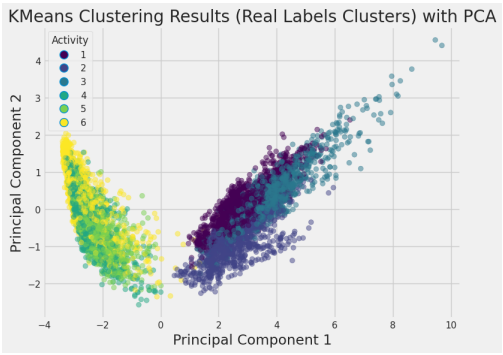

Figure 8: KMeans Visualization PCA (Real Labels)

   (b) Min Samples: This defines the minimum number of data points required to form a dense region or cluster. Specifically, it determines the density threshold necessary for a point to be considered a core point, which is fundamental to the DBSCAN algorithm. Adjusting the min samples parameter impacts the granularity of the resulting clusters: higher values yield larger, more sparse clusters, while lower values produce smaller, denser clusters.

To find the optimal value for the epsilon - **5.86**, the elbow method using the KNearest Neighbour was employed:

The min sample sample was choosen from a pool of likely candidates by comparing the Silhouette and Davies-Bouldin scores.

The results of this clustering are shown below:

After using the elbow method to find the optimal epsilon value and using the Silhoutte and Davies

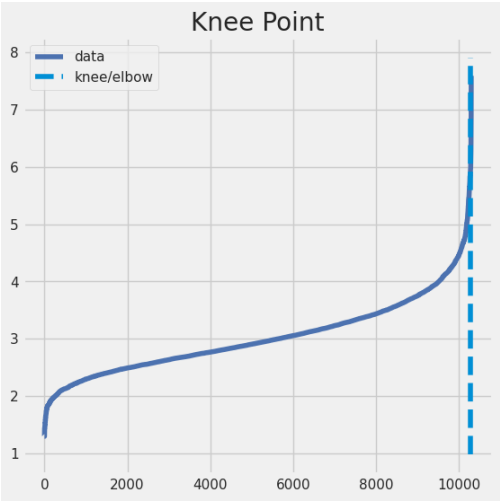

Figure 9: DBSCAN optimal epsilon

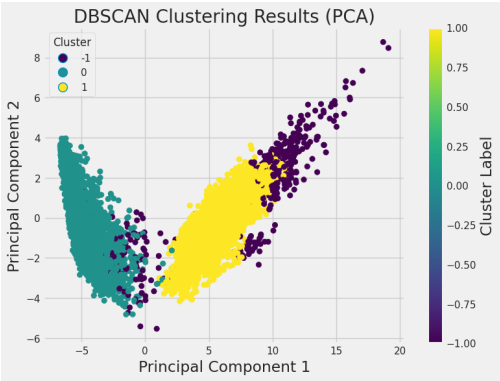

Figure 10: DBSCAN Visualization PCA

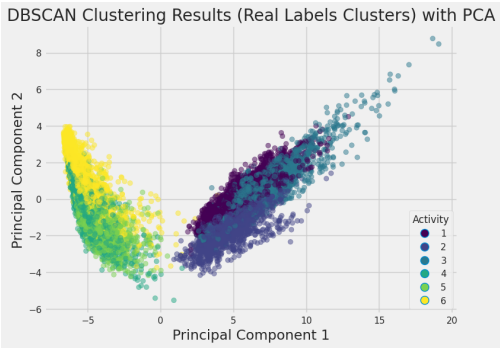

Figure 11: DBSCAN Visualization PCA (Real Labels

Bouldin score to extrapolate the optimal min samples values, the DBSCAN algorithm classified the data into 3 clusters contrary to the 2 clusters provided by the KMeans algorithm.

From the diagram, we can see that DBSCAN separates the data into two clusters 0 and 1 with the noise cluster being identified as -1. This is due to DBSCAN's sensitivity to noise and it's algorithm assigning noises to a separate cluster.

3.2. Dimensionality Reduction

In the second part of the project, dimensionality reduction was applied on the dataset before performing clustering.

As we've seen above, the data's high dimensionality introduces difficulties for the clustering algorithms. In terms of visualization, despite using RandomForest to choose the most important features to highlight the clustering performed, a dimensionality reduction technique PCA had to be employed to visualize the data in a presentable manner.

Dimensionality reduction is an important technique in data analysis and machine learning aimed at reducing the number of features or variables in a dataset while preserving its essential characteristics. It addresses the "curse of dimensionality" by simplifying high-dimensional data into a lower-dimensional space, making it more manageable, interpretable, and computationally efficient. Dimensionality reduction methods can broadly be categorized into two types: feature selection and feature extraction. In our project, we use feature extraction methods, that can create new features that are combinations of the original features, typically through techniques like Principal Component Analysis (PCA), Singular Value Decompositon (SVD) or LDA . These methods aim to capture the most relevant information in the data while reducing redundancy and noise.

The LDA with 3 components is selected for this project. Linear Discriminant Analysis (LDA) is a dimensionality reduction technique commonly used for clustering tasks. Unlike other methods such as PCA, LDA considers class information, aiming to find the feature subspace that maximizes class separability while minimizing intra-class variance. By projecting high-dimensional data onto a lower-dimensional space, LDA extracts discriminative features that facilitate clustering by grouping similar data points together. This reduction in dimensionality not only

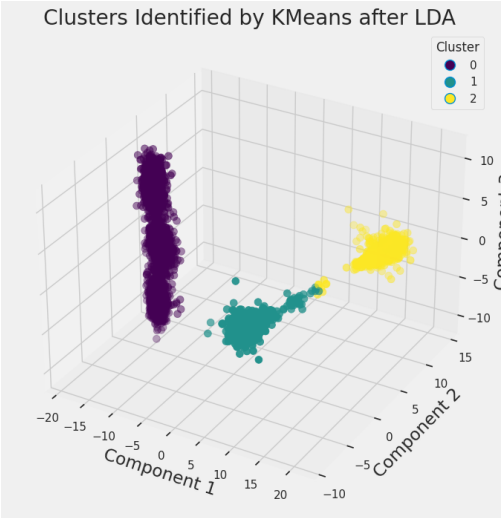

Figure 12: KMeans Visualization - LDA

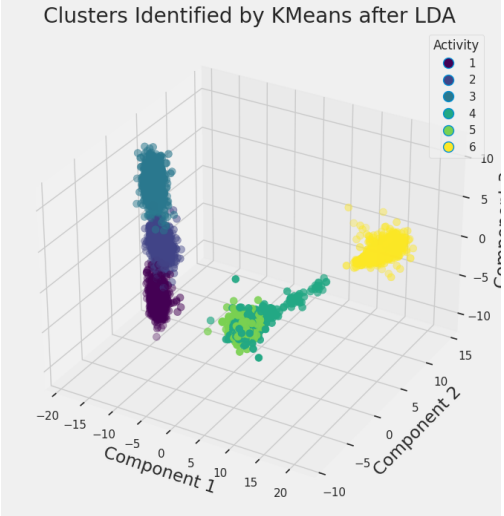

Figure 13: KMeans Visualization - LDA (Real Labels)

simplifies the clustering process but also enhances its effectiveness by preserving class-specific information, making LDA a valuable tool in various data clustering applications.

1. KMeans: In the KMeans model, after applying dimensionality reduction, the optimal clusters identified changed to **3**. This is due to the ability of the model to better capture differences between the data that it previously couldn't.
   The results are shown in Figure 12 and Figure 13:
2. DBSCAN: After applying dimensionality reduction, the optimal value of epsilon using the knee method became **1.79** and then the elbow method for the Silhouette and Davies Bouldin scores was used to further validate the result as shown in Figure 14 and Figure 15.
   Similar to the KMeans results, applying dimensionality reduction on the DBSCAN facilitated the creation of better and more informative clusters. One difference is the Noise cluster (label -1) present in the DBSCAN that represents data points considered outliers due to the DBSCAN's sensitivity to noise.
   The results are shown in Figure 16 and Figure 17
   After using the elbow method to find the optimal epsilon value and using the Silhoutte and Davies Bouldin score to extrapolate the optimal min samples

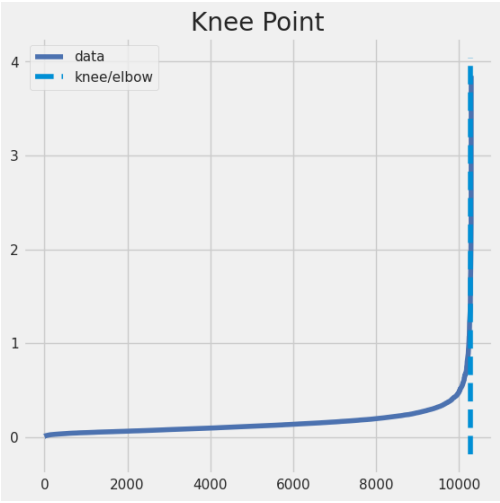

Figure 14: DBSCAN - LDA Knee

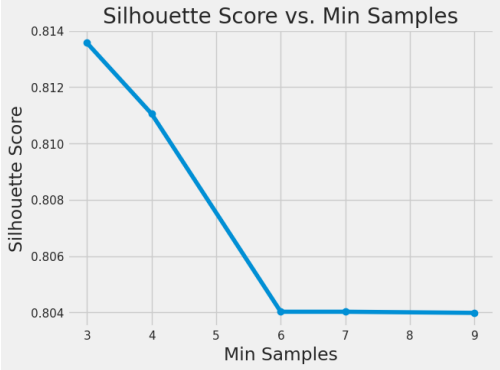

Figure 15: DBSCAN - LDA Elbow

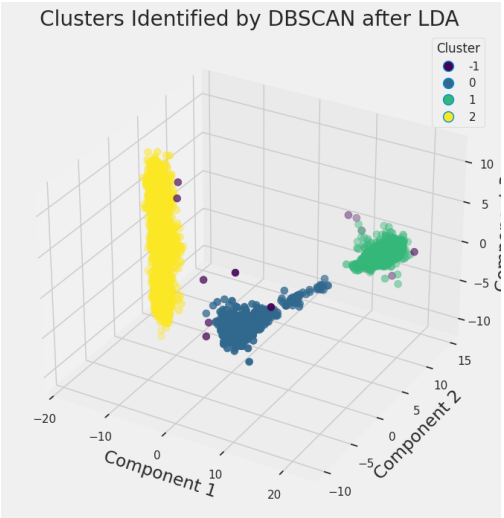

Figure 16: DBSCAN Visualization LDA

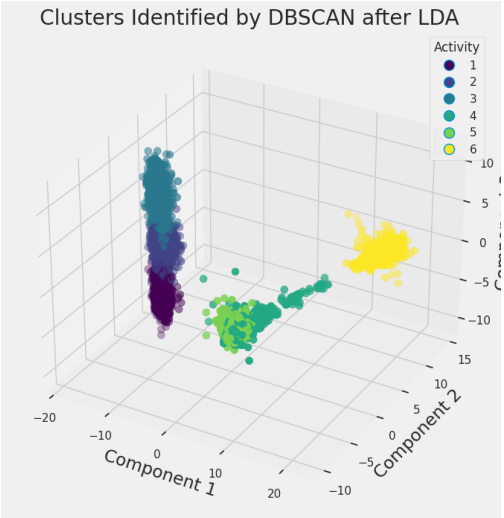

Figure 17: DBSCAN Visualization LDA (Real Labels)

values, the DBSCAN algorithm classified the data into 3 clusters contrary to the 2 clusters provided by the KMeans algorithm.
From the diagram, we can see that DBSCAN separates the data into two clusters 0 and 1 with the noise cluster being identified as -1. This is due to DBSCAN's sensitivity to noise and it's algorithm assigning noises to a separate cluster.

4. Discussion

In the first iteration of the project, the KMeans and DBSCAN models were trained without employing dimensionality reduction. For the KMeans method, the elbow method together with the metrics used: Silhoutte score and Davies-Bouldin score, the optimal value of 2 clusters was found. The algorithm was able to effectively group the data points into two representative clusters. However, the complexity of the data made visualization tedious despite employing a feature selection algorithm to select the most pertinent features of the dataset. The PCA dimensionality reduction technique had to be employed to visualize the data in a more presentable manner.

Similarly, the elbow method was used for the DBSCAN to find the optimal epsilon value. Then using the domain knowledge, trail and error, and possible min-value

parameters, the model was fit on the data and clusters were formed. DBSCAN sensitivity to noise and the data's high dimensionality made clustering challenging and the resulting visualization as well.

In the second part, the LDA dimensionality reduction technique was applied on the data before clustering. This simplified the highly dimensional data into 3 components facilitating the subsequent fitting of clustering models. The results showed better metric performance, clustering results and more informative/representative clustering visualizations.

One important thing to also notice is, from the graphs after applying dimensionality reduction with the plot of the real labels as shown above, we can see that the alogrithm grouped the ACTIVE tasks of [1', 'WALKING'], ['2', 'WALKING-UPSTAIRS'], ['3', 'WALKING-DOWNSTAIRS'] into the 0 cluster and separated the PASSIVE tasks into two: ['4', 'SITTING'], ['5', 'STANDING'] into cluster 1 and ['6', 'LAYING'] as a separate cluster 2. Despite labels 4, 5, and 6 being PASSIVE tasks, the algorithm could further differentiate between by considering the angles and position differences the two tasks entail.

Table 1: Comparison of Model Silhoutte Scores

Model	**Base Model**	**Dimensionality Reduction**
KMeans	0.48	0.82
DBSCAN	0.41	0.81

Table 2: Comparison of Model Davies Bouldin Scores

Model	**Base Model**	**Dimensionality Reduction**
KMeans	0.84	0.23
DBSCAN	2.62	1.05

Table 3: Comparison of Model Execution Time

Model	**Base Model**	**Dimensionality Reduction**
KMeans	1.08	0.085
DBSCAN	9.62	0.622

Also from the graphs above, we could the performance improvement we got from using dimensionality reduction. The models silhoutte scores got higher, the Davies Bouldin Scores got lower and the execution time also become significantly less.

In conclusion, from this project, we can deduce the importance of carefully choosing hyperparameters for the complex task of unsupervised learning and the importance of dimensionality reduction techniquess especially for high dimensional data in easing the clustering, improving performance of models and simplifying the visualization processes.

5. Conclusion

In this project, we began by analyzing the data and then performning clustering using two well-known clustering techniques without using dimensionality reducition. We then applied dimensionality reduction and the model's improved in all the measured aspects.