# Mini Project 3 : Clustering Human Activity

Harry Setiawan Hamjaya

2302000

***Abstract*— In this project, we delve into clustering method using the Human Activity Recognition Using Smartphones dataset, aiming to construct and evaluate two unsupervised machine learning models for clustering the human activities. Leveraging traditional yet powerful methodologies such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN) and K-Means, we rigorously experiment with model, hyperparameter tuning, and dimensional reduction with t-Distributed Stochastic Neighbor Embedding (t-SNE) to cluster human activities from approximately 10,300 rows of dataset. Our findings highlight the strengths and limitations of each approach, showcasing the importance of adaptability and innovation in navigating the dynamic landscape of clustering with dimension reduction approach. Through this exploration, we contribute valuable insights and implications for the advancement of unsupervised machine learning method especially clustering applications.**

**Keywords: Clustering, DBSCAN, K-Means, t-SNE, and Human Activity Recognition Using Smartphones.**

## I. Introduction

In today's era of ubiquitous technology, the integration of smartphones into everyday life has become pervasive. With the advancement of sensor technology, smartphones now serve as powerful tools for gathering data about human movement and behavior. In this context, our project delves into the realm of activity recognition using sensor data collected from smartphones.

The problem we aim to address revolves around accurately identifying and categorizing various physical activities performed by individuals, such as walking, walking upstairs, walking downstairs, sitting, standing, and lying down. This classification task is crucial for numerous applications, ranging from healthcare monitoring to personalized fitness tracking and beyond.

Our dataset comprises recordings from 30 volunteers, each wearing a smartphone equipped with an accelerometer and gyroscope. These sensors capture 3-axial linear acceleration and 3-axial angular velocity at a consistent rate of 50Hz. To enhance data quality, noise filters were applied during preprocessing, followed by segmentation into fixed-width sliding windows with a 50% overlap. Additionally, a Butterworth low-pass filter was utilized to separate gravitational and body motion components, aiding in feature extraction.

The challenge lies in effectively analyzing these sensor signals to extract meaningful features that capture the nuances of different activities. Furthermore, the dataset has been partitioned into training and test sets, enabling robust model evaluation and validation.

Through this project, we aim to develop machine learning models capable of accurately classifying human activities based on smartphone sensor data. By doing so, we not only contribute to advancing activity recognition technology but also pave the way for practical applications that benefit from automated monitoring and analysis of human behavior.

## II. Exploratory Data Analysis

### A. *Label*

The dataset label represents the distribution of different activities performed by individuals in the dataset. Each activity is labeled, and the numbers next to the labels indicate the frequency or count of occurrences of each activity within the dataset. Here's a breakdown of the labels and their corresponding counts:

LAYING: This label indicates that the individual was lying down. The count associated with it (1944) suggests that this activity occurred 1944 times within the dataset.

STANDING: Refers to the activity of standing upright. It occurred 1906 times.

SITTING: Indicates the activity of sitting down. This activity occurred 1777 times.

WALKING: Represents the activity of walking on a level surface. It occurred 1722 times.

WALKING_UPSTAIRS: Describes the activity of walking upstairs. This activity occurred 1544 times.

WALKING_DOWNSTAIRS: Denotes the activity of walking downstairs. It occurred 1406 times.

These labels and their corresponding counts provide insights into the distribution of activities recorded within the dataset as depicted in figure 1 is crucial for understanding the composition of the dataset and can guide the development and evaluation of activity recognition models.

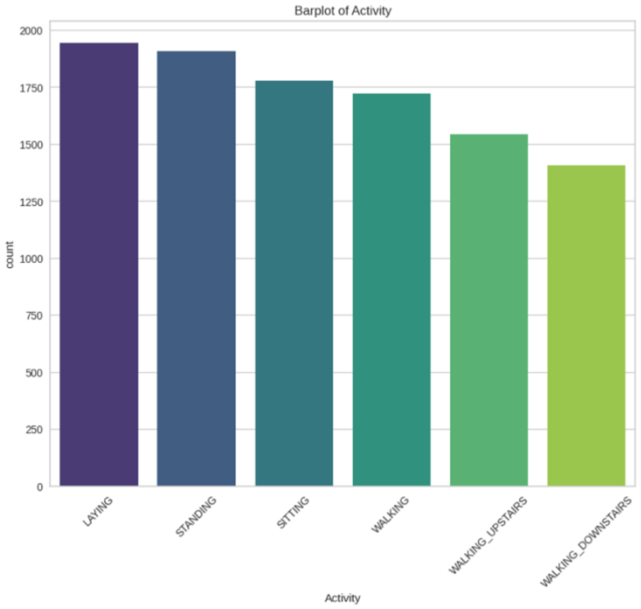


Figure 1: The Distribution of Class Label

### B. *Feature*

The dataset features represent various aspects of the sensor data collected during the experiment, which include both time and frequency domain variables extracted from accelerometer and gyroscope signals. Here's an explanation of the features provided:

1. fBodyAcc, fBodyGyro, fBodyAccJerk: These features represent frequency domain variables derived from the Fourier Transform of the accelerometer, gyroscope, and accelerometer jerk signals, respectively. Each category includes 79 different features.
2. tGravityAcc, tBodyAcc, tBodyGyroJerk, tBodyGyro: These features represent time domain variables of the accelerometer, body acceleration, gyroscope jerk, and gyroscope signals, respectively. Each category includes 40 different features.
3. tBodyAccMag, tGravityAccMag, tBodyAccJerkMag, tBodyGyroMag, tBodyGyroJerkMag, fBodyAccMag, fBodyBodyAccJerkMag, fBodyBodyGyroMag, fBodyBodyGyroJerkMag: These features represent magnitude calculations in the time and frequency domains for the accelerometer, gravity acceleration, accelerometer jerk, gyroscope, gyroscope jerk, and their respective Fourier Transform. Each category includes 13 different features.
4. angle: This category includes 7 features representing angles between vectors.
5. subject: This feature identifies the subject who carried out the experiment. It is an identifier for each individual participant in the study.
6. type: This feature serves as the type of the dataset for the activity being performed during the data recording. It represents either train or test.

Overall, these features provide a comprehensive representation of the sensor data collected during the experiments, covering both the time and frequency domains, along with various magnitude and angle calculations. They are essential for building models to classify activities based on the recorded sensor data.

Table 1. Dataset Features

| Feature type | count |
|---|---|
| fBodyAcc | 79 |
| fBodyGyro | 79 |
| fBodyAccJerk | 79 |
| tGravityAcc | 40 |
| tBodyAcc | 40 |
| tBodyGyroJerk | 40 |
| tBodyGyro | 40 |
| tBodyAccJerk | 40 |
| tBodyAccMag | 13 |
| tGravityAccMag | 13 |
| tBodyAccJerkMag | 13 |
| tBodyGyroMag | 13 |
| tBodyGyroJerkMag | 13 |
| fBodyAccMag | 13 |
| fBodyBodyAccJerkMag | 13 |
| fBodyBodyGyroMag | 13 |
| fBodyBodyGyroJerkMag | 13 |
| angle | 7 |
| subject | 1 |
| type | 1 |

## III. DATA PROCESSING

### A. *Merging*

In the preprocessing phase of our dataset, two primary steps were undertaken. Firstly, we merged the train and test datasets, a decision made to enrich the data available for clustering. By consolidating both sets, we created a larger and more diverse dataset, which is advantageous for unsupervised methods like clustering. Since clustering doesn't require labeled data for training, we found it unnecessary to include the test set data for this analysis. Moreover, merging the datasets simplified the preprocessing pipeline, ensuring uniform treatment of the entire dataset and consistency in data transformations and clustering procedures.

### B. *Scaling*

Secondly, we scaled the dataset using a standard scaler. This common preprocessing technique standardizes features by removing the mean and scaling to unit variance. Scaling is typically applied to ensure that all features contribute equally to the clustering process and to mitigate the influence of features with larger scales. It aids in improving the convergence of clustering algorithms and enhances their performance. However, upon observation, we found that scaling the dataset didn't significantly improve the clustering results. As a result, we opted to drop the scaling step from the final preprocessing pipeline.

## IV. MODELLING

In approaching the clustering task using K-Means and DBSCAN algorithms on the given dataset, several crucial aspects were considered to ensure effective clustering results. Here's an explanation of the key considerations and methodologies employed:

1. Choosing the Number of Clusters:

   For K-Means: The number of clusters was determined using the Elbow method and silhouette score. The Elbow method helps to identify the point where the within-cluster sum of squares (WCSS) starts to decrease at a slower rate, indicating the optimal number of clusters. Additionally, the silhouette score measures the compactness and separation between clusters, with higher scores indicating better-defined clusters. These methods suggested at least 2 clusters for the original dataset as shown in figure 2.

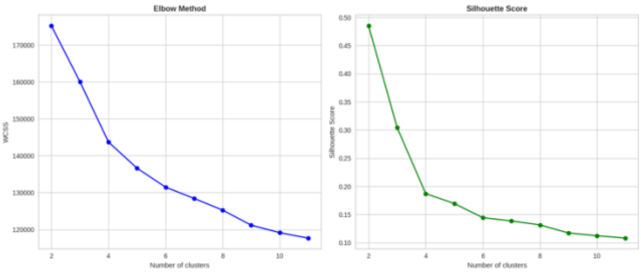


Figure 2: The Elbow Method and Silhouette Score

For DBSCAN: In the context of clustering algorithms, particularly DBSCAN (Density-Based Spatial Clustering of Applications with Noise), the parameter "k" refers to the number of nearest neighbors considered when determining the core points and density of clusters. For DBSCAN, each data point is evaluated based on its proximity to other points within a specified distance epsilon (ε). The number of nearest neighbors, denoted as "k," determines how many neighboring points are considered when evaluating the density around each data point. When setting the number of nearest neighbors to k=50, it means that for each data point, the algorithm will consider the 50 closest neighboring points within a radius defined by epsilon as shown in figure 3, the epsilon would be around 6.

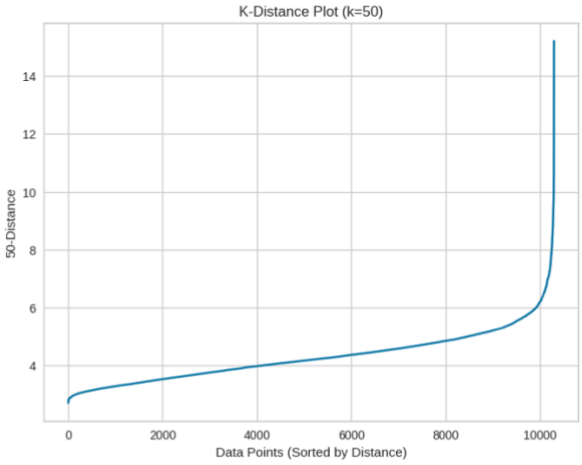
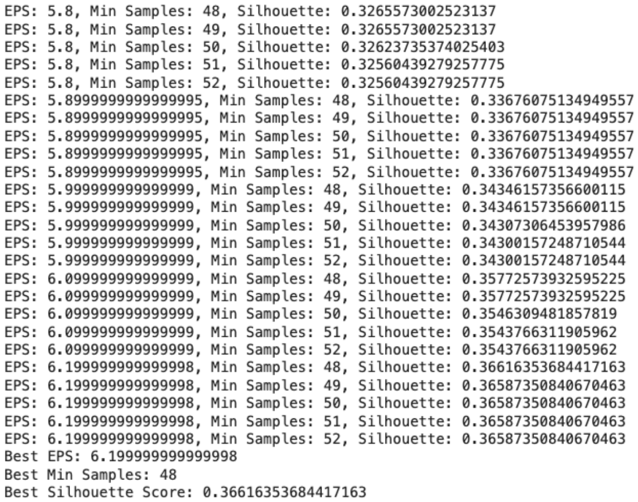


Figure 3: The K-Distance Method

2. Finding Optimal Parameter Values:

For K-Means: The optimal number of clusters was identified using the Elbow method and silhouette score, providing insights into the dataset's inherent structure and the appropriate number of clusters to represent it effectively.

For DBSCAN: Grid search and Optuna were used to find optimal parameter values such as epsilon based on the given minimum samples as shown in figure 4. By systematically exploring a range of parameter values and evaluating their impact on clustering performance, the most suitable parameters for DBSCAN were determined.



```
EPS: 5.8, Min Samples: 48, Silhouette: 0.3265573002523137
EPS: 5.8, Min Samples: 49, Silhouette: 0.3265573002523137
EPS: 5.8, Min Samples: 50, Silhouette: 0.32623735374025403
EPS: 5.8, Min Samples: 51, Silhouette: 0.32560439279257775
EPS: 5.8, Min Samples: 52, Silhouette: 0.32560439279257775
EPS: 5.8999999999999995, Min Samples: 48, Silhouette: 0.33676075134949557
EPS: 5.8999999999999995, Min Samples: 49, Silhouette: 0.33676075134949557
EPS: 5.8999999999999995, Min Samples: 50, Silhouette: 0.33676075134949557
EPS: 5.8999999999999995, Min Samples: 51, Silhouette: 0.33676075134949557
EPS: 5.8999999999999995, Min Samples: 52, Silhouette: 0.33676075134949557
EPS: 5.999999999999999, Min Samples: 48, Silhouette: 0.34346157356600115
EPS: 5.999999999999999, Min Samples: 49, Silhouette: 0.34346157356600115
EPS: 5.999999999999999, Min Samples: 50, Silhouette: 0.34307306453957986
EPS: 5.999999999999999, Min Samples: 51, Silhouette: 0.34300157248710544
EPS: 5.999999999999999, Min Samples: 52, Silhouette: 0.34300157248710544
EPS: 6.099999999999999, Min Samples: 48, Silhouette: 0.35772573932595225
EPS: 6.099999999999999, Min Samples: 49, Silhouette: 0.35772573932595225
EPS: 6.099999999999999, Min Samples: 50, Silhouette: 0.3546309481857819
EPS: 6.099999999999999, Min Samples: 51, Silhouette: 0.3543766311905962
EPS: 6.099999999999999, Min Samples: 52, Silhouette: 0.3543766311905962
EPS: 6.199999999999998, Min Samples: 48, Silhouette: 0.36616353684417163
EPS: 6.199999999999998, Min Samples: 49, Silhouette: 0.36587350840670463
EPS: 6.199999999999998, Min Samples: 50, Silhouette: 0.36587350840670463
EPS: 6.199999999999998, Min Samples: 51, Silhouette: 0.36587350840670463
EPS: 6.199999999999998, Min Samples: 52, Silhouette: 0.36587350840670463
Best EPS: 6.199999999999998
Best Min Samples: 48
Best Silhouette Score: 0.36616353684417163
```

Figure 4: The K-Distance Method

3. Data Processing Steps:

The decision was made to keep the original dataset without standard scaling. Despite attempts to scale the data using a standard scaler, it was observed that the dimensionality remained too large for effective visualization. Therefore, the scaling step was dropped from the preprocessing pipeline.

Additionally, unnecessary features or dimensions that didn't contribute significantly to the clustering process were dropped to simplify the dataset and improve computational efficiency.

In summary, the clustering process involved thoughtful consideration of the number of clusters, optimal parameter values, and appropriate data processing steps for both K-Means and DBSCAN algorithms. While K-Means relied on the Elbow method and silhouette score, DBSCAN utilized grid search and Optuna for parameter optimization. The decision to keep the original dataset and drop unnecessary features was based on observations regarding the effectiveness of data scaling and the dimensionality of the dataset.

## V. Dimensional Reduction

A dimensionality reduction technique was applied before using K-Means and DBSCAN on the dataset. Here's an explanation of the key aspects and observations:

1. Dimensionality Reduction Technique:

Both t-SNE (t-distributed Stochastic Neighbor Embedding) and PCA (Principal Component Analysis) were considered for dimensionality reduction. PCA is known for its computational efficiency, while t-SNE is favored for its ability to preserve local relationships and visualize high-dimensional data in lower dimensions.

After visualizing both techniques in figure 5 and figure 6 respectively, t-SNE was chosen due to its superior ability to capture the spread between clusters, providing clearer insights into the dataset's structure.

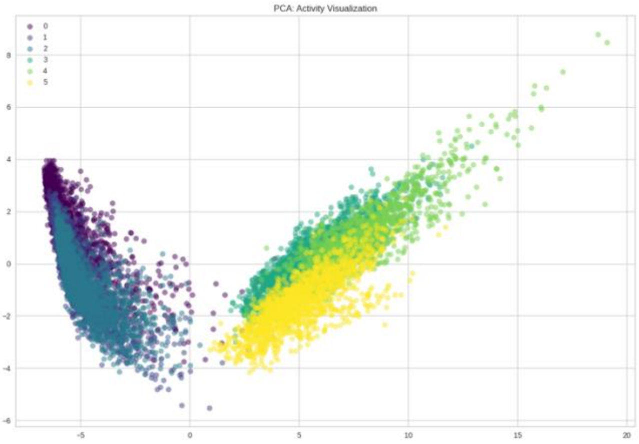


Figure 5: PCA Plot

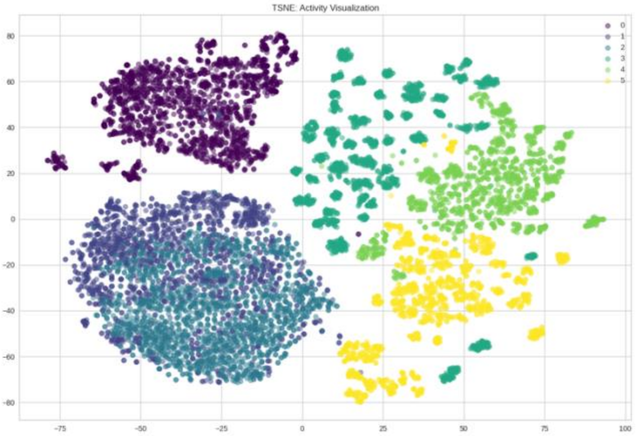


Figure 6: t-SNE plot

2. Effect on Code Efficiency:

Dimensionality reduction significantly improved code efficiency in terms of computational resources. With the original dataset containing 561 features, computations for clustering could be computationally expensive and time-consuming. By reducing the dimensionality using t-SNE, the number of features was greatly reduced, resulting in faster computations and improved efficiency.

One of the most significant effects of the dimensional reduction is the elbow method and silhouette score with K-Means since it changes the number of clusters shown in figure 7.

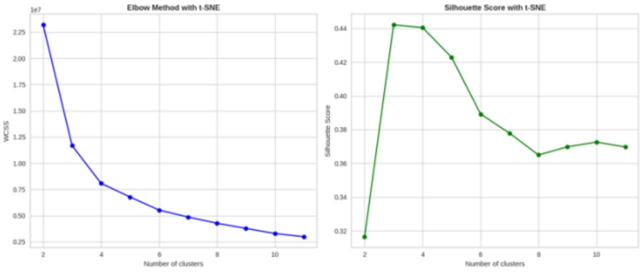


Figure 7: K-Means after t-SNE

On the other hand, the result toward DBSCAN is not that significant since instead of the cluster DBSCAN method is exploting the possible parameter such as epsilon which make sense that it might be similar as shown in figure 8, but there is an improvement of epsilon from 6 before t-SNE to 8.

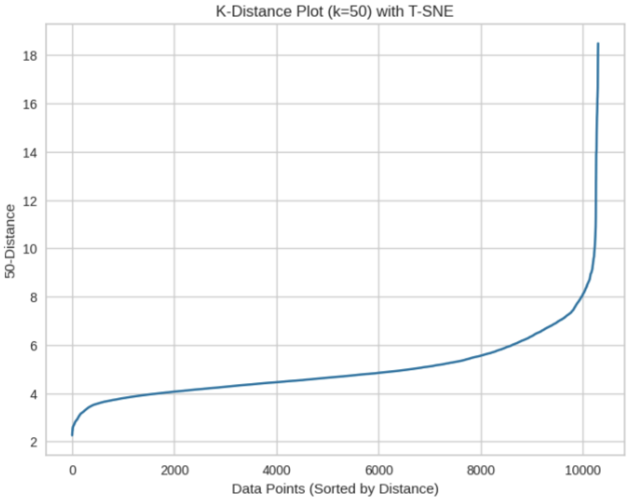


Figure 8: The K-Distance Method

The most significant improvement can be shown by the comparison between using 2 features (after t-SNE) versus 561 features (original dataset) illustrates the drastic reduction in computational complexity achieved through dimensionality reduction as shown in table 2. Based on table 2, DBSCAN improve around 5.4 seconds during the clustering, while K-Means improve around 0.18 seconds

Table 2. t-SNE time performance comparison

| Before t-SNE is implemented | |
|---|---|
| Algorithm | Time Performance |
| DBSCAN | 5.551645040512085 seconds |
| K-Means | 0.1993107795715332 seconds |

| After t-SNE is implemented | |
|---|---|
| Algorithm | Time Performance |
| DBSCAN | 0.12137484550476074 seconds |
| K-Means | 0.017981529235839844 seconds |

3. Comparison of Clustering Outcome:

The comparison presented in Table 3 showcases the impact of implementing t-SNE (t-distributed Stochastic Neighbor Embedding) on the clustering performance of two algorithms: DBSCAN and K-Means. Initially, before the application of t-SNE, DBSCAN identified only one cluster, indicating that the algorithm struggled to discern meaningful groupings within the dataset. Similarly, K-Means identified two clusters, suggesting a limitation in its ability to capture the underlying structure of the data effectively. However, following the implementation of t-SNE, significant improvements were observed in both algorithms' clustering outcomes. DBSCAN, after t-SNE, successfully identified six distinct clusters, indicating a more nuanced understanding of the data's underlying patterns and densities. Similarly, K-Means now identified four clusters, showcasing a more refined segmentation of the dataset compared to its performance before t-SNE. This transformation highlights the effectiveness of t-SNE in reducing the dataset's dimensionality while preserving the local relationships between data points, ultimately leading to enhanced clustering performance and a more accurate representation of the dataset's structure.

Table 3. t-SNE time performance comparison

| Before t-SNE is implemented | |
|---|---|
| Algorithm | Number of Cluster |
| DBSCAN | 1 |
| K-Means | 2 |

| After t-SNE is implemented | |
|---|---|
| Algorithm | Number of Cluster |
| DBSCAN | 6 |
| K-Means | 4 |

In summary, the implementation of t-SNE as a dimensionality reduction technique greatly enhanced code efficiency by reducing computational complexity. Additionally, it improved the clustering outcome by providing clearer insights into the dataset's structure and enabling clustering algorithms to identify more meaningful clusters.

## VI. VISUALIZATION

Before the implementation of dimensionality reduction techniques such as t-SNE (t-distributed Stochastic Neighbor Embedding), visualizing the dataset posed significant challenges due to its high dimensionality. With 561 features, the dataset represented a complex and multi-dimensional space, making it difficult to comprehend and visualize directly. The sheer number of dimensions made it impractical to plot the data in a comprehensible manner, hindering any meaningful interpretation of its structure or patterns. As a result, understanding the relationships and distributions within the dataset proved to be a formidable task.

However, after applying t-SNE, the dataset underwent a transformation, reducing its dimensionality while preserving the local relationships between data points. This transformation allowed the dataset to be represented in a lower-dimensional space, typically two or three dimensions, making it feasible to visualize. By condensing the dataset into a more manageable space, t-SNE facilitated the visualization of the dataset's structure and patterns. Consequently, insights into the data's inherent clusters, densities, and relationships became accessible, enabling a deeper understanding of its underlying characteristics.

Figure 9 and 10 depict the results of DBSCAN clustering applied to the dataset after dimensionality reduction using t-SNE, with the tuning of parameters performed through Grid Search in Figure 9 and Optuna in Figure 10. In both figures, the clustering outcomes are visually represented, providing insights into the effectiveness of the clustering algorithms in partitioning the data into distinct clusters. The use of t-SNE for dimensionality reduction facilitated a more intuitive visualization of the dataset, enabling clearer delineation of clusters in lower-dimensional space. In Figure 9, where Grid Search was employed for parameter tuning, the clustering result showcases the identification of several well-defined clusters, each represented by distinct colors or markers. Similarly, in Figure 10, where parameter optimization was conducted using Optuna, the clustering outcome reveals the successful partitioning of the data into separate clusters, albeit with potentially different cluster configurations compared to Figure 9. Overall, both figures demonstrate the efficacy of DBSCAN clustering in capturing the underlying structure of the dataset, with t-SNE playing a pivotal role in enabling meaningful visualization and interpretation of the clustering results.

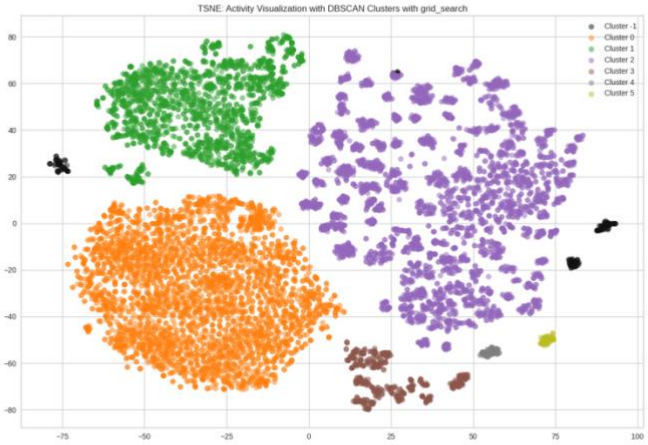


Figure 9: DBSCAN with t-SNE and Grid Search

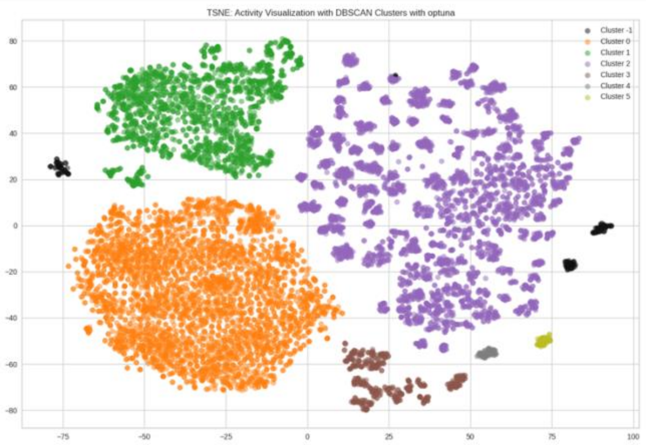


Figure 10: DBSCAN with t-SNE and Optuna

Figures 11 and 12 offer insights into the K-Means clustering analysis conducted on the dataset following dimensionality reduction using t-SNE. In Figure 11, the clustering outcome illustrates the result of applying K-Means with a configuration set to identify three distinct clusters. On the other hand, Figure 12 showcases the clustering outcome with K-Means configured to identify four clusters, with silhouette scores of 0.44212314 and 0.44045177, respectively. Silhouette score serves as a metric for evaluating the cohesion and separation of clusters, with higher scores indicating better-defined clusters. Both figures present a visual representation of the clustering results, where data points belonging to different clusters are visually distinguished by unique colors or markers. The utilization of t-SNE for dimensionality reduction allows for the visualization of the dataset in lower-dimensional space, facilitating the interpretation of clustering outcomes. In Figure 11, where three clusters are identified, we observe a discernible partitioning of the data into coherent clusters. Similarly, Figure 12, with the identification of four clusters based on high silhouette scores, reveals a more refined segmentation of the dataset, indicating the algorithm's ability to capture finer distinctions within the data. Overall, Figures 11 and 12 demonstrate the effectiveness of K-Means clustering in segmenting the dataset into meaningful clusters, with t-SNE playing a crucial role in enabling visualization and interpretation of clustering results.

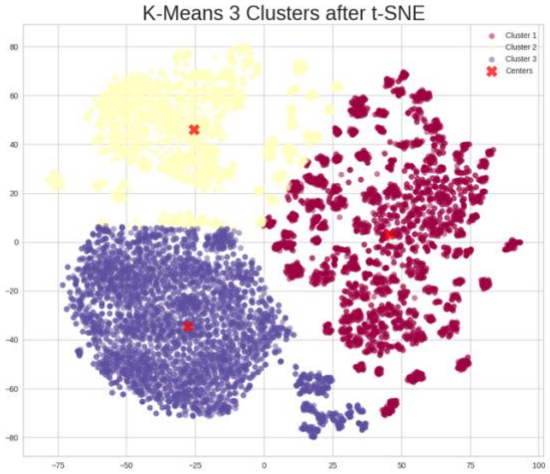


Figure 11: K-Means 3 Clusters

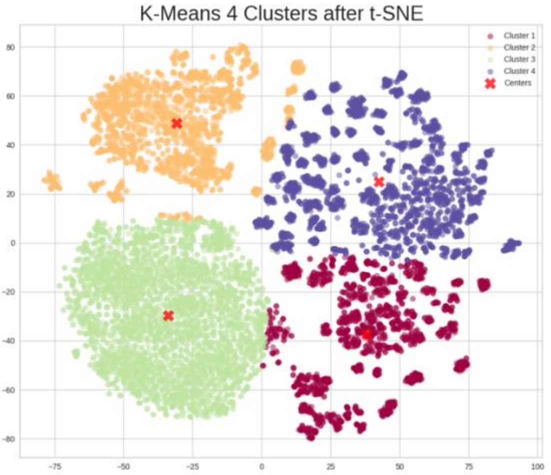


Figure 12: K-Means 4 Clusters

In essence, while visualizing the dataset before dimensional reduction was challenging due to its high dimensionality, the implementation of t-SNE made visualization feasible by reducing the dataset's dimensionality while retaining important relationships between data points. This transformation empowered analysts to explore and interpret the dataset more effectively, ultimately leading to improved insights and decision-making.

## VII. Conclusion

In conclusion, the identified clusters represent distinct groupings within the dataset based on similarities in the data points' characteristics. Each cluster may correspond to different patterns, behaviours, or attributes present in the dataset, depending on the specific context of the analysis. For example, in the context of activity recognition based on smartphone sensor data, clusters may represent different types of activities performed by individuals, such as laying, standing and walking.

The interpretation of identified clusters requires a deeper understanding of the dataset and domain knowledge. By analysing the characteristics and patterns within each cluster, we can infer the underlying factors contributing to the cluster formation. This interpretation provides valuable insights into the dataset's structure and aids in drawing meaningful conclusions or making data-driven decisions. For example it’s pretty clear that in both K-Means and DBSCAN the “Laying” and “Standing” will be in the opposite sides of the “Walking” state as visualized by the t-SNE algorithm.

Throughout the analysis, several scientific bottlenecks may have been encountered. These could include challenges related to data pre-processing since the way we tried with the standard scaler didn’t improve the model, algorithm selection whether should we approach with the unsupervised machine learning approach or supervised method instead, and interpretation of clustering results would be hard because need more domain knowledge to understand how we should cluster the dataset. Overcoming these bottlenecks often requires a combination of domain expertise, methodological rigor, and experimentation.

There are several ways to overcome these problems. Firstly thorough Data Understanding by ensuring a comprehensive understanding of the dataset and its characteristics and domain knowledge is essential for effective analysis and interpretation. Another approach is through more experimentation and iteration by iteratively experimenting with different techniques, algorithms, and parameter settings allows for refinement and improvement of results over time. Finally, collaboration and peer review for seeking input and feedback from domain experts and peers can provide valuable insights and help address potential blind spots or biases in the analysis. By leveraging these strategies, we can overcome scientific bottlenecks and conduct a rigorous and insightful analysis that yields meaningful interpretations of identified clusters and their implications.