Machine Learning, 2024

Mini Project 3

Clustering of Human Activity using KMeans and DBSCAN

Project Report

Author: Halidu Abdulai

Student ID: 2301998

Contents

1 Introduction

In the current era of pervasive computing, the integration of smart devices into our daily lives has unlocked new frontiers in understanding human behavior through data analysis. The project at hand leverages this technological integration, focusing on the Human Activity Recognition (HAR) domain using data collected from smartphones. This endeavor utilizes a dataset derived from experiments involving 30 volunteers, spanning a diverse age group, who carried out six distinct activities while equipped with Samsung Galaxy S II smartphones. The devices' embedded sensors, including accelerometers and gyroscopes, captured a rich stream of data regarding 3-axial linear acceleration and 3-axial angular velocity, facilitating a comprehensive analysis of human movement.

According to the documentation of the dataset provided, the dataset's preparation involved meticulous pre-processing steps to ensure data integrity and relevance, highlighting the challenges and considerations in handling real-world sensor data. This included applying noise filters and segmenting the data using fixed-width sliding windows, a testament to the complexity of processing sensor signals for meaningful analysis.

This project's primary objective is to employ clustering techniques, specifically K-Means and DBSCAN, to unravel the underlying patterns within this dataset, providing insights into the nature of human activities captured through smartphone sensors. This involves critical decision-making processes regarding the selection of the number of clusters, determination of optimal parameters, and the implementation of dimensionality reduction techniques to enhance computational efficiency and clustering accuracy.

This report aims to present a detailed account of the methodologies adopted, from data processing to model implementation, culminating in a scientific discussion on the findings. Through this investigation, the I seek not only to contribute to the advancement of HAR techniques but also to reflect on the broader implications of utilizing smart devices in understanding human dynamics, encapsulating the blend of technology and human behavior analysis.

2 Methodology

Several steps were taken to obtain the necessary results. Some of the paramount steps undertaken are outlined below.

2.1 Data loading and preprocessing

The UCI HAR dataset made available for this project came in an unprocessed state. The dataset was split across several text files. To ensure that the dataset was easy to work with, a function was written to load the data from the different text files, aggregate the resulting data into a single CSV file, each for the train and test datasets. The dataset was then checked for null values. Fortunately, the dataset had no null values. In addition, the dataset had already been normalized which left little work to be done on the preprocessing step.

The distribution of the train/test datasets according to the target variable (activity) was checked and the results are shown in the following two figures.

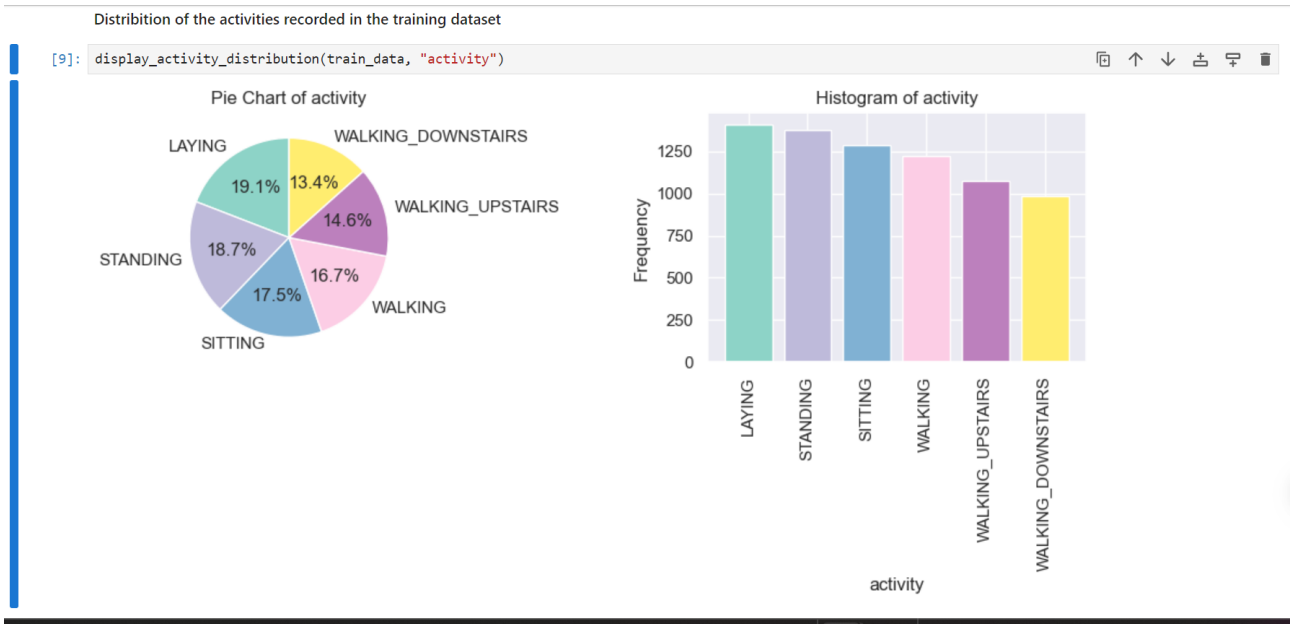


Figure 1: Distribution of the activities recorded in the train dataset

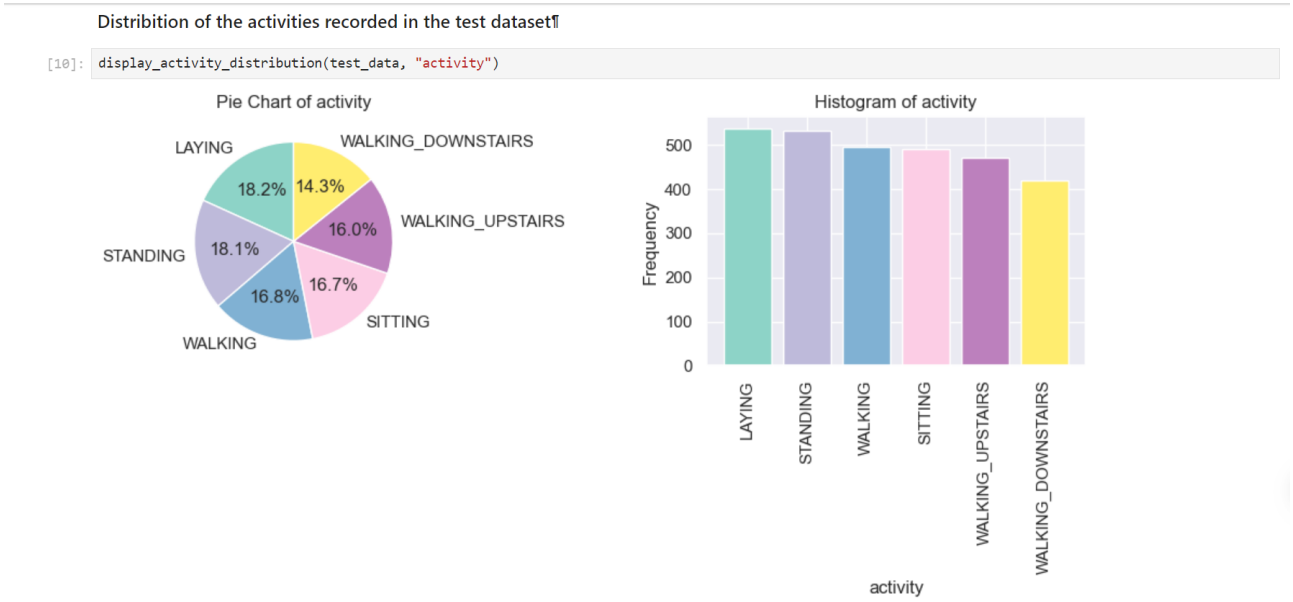


Figure 2: Distribution of the activities recorded in the test dataset

2.2 Modelling/Performing Clustering

The main goal of this project was to use KMeans and DBSCAN to perform clustering to find out the different clusters available in the dataset. For this, different techniques such finding the best k values for the KMeans and, epsilon and minimum samples parameters for the DBSCAN were employed. Next, I summarize the main tasks performed during the clustering.

1. **KMeans Clustering**: To run the KMeans, the number of clusters need to be predetermined. To achieve this, I employed the elbow and silhoutte analysis methods to determine the best k value. This value was found by both methods to be 2 as illustrated in the figures below.

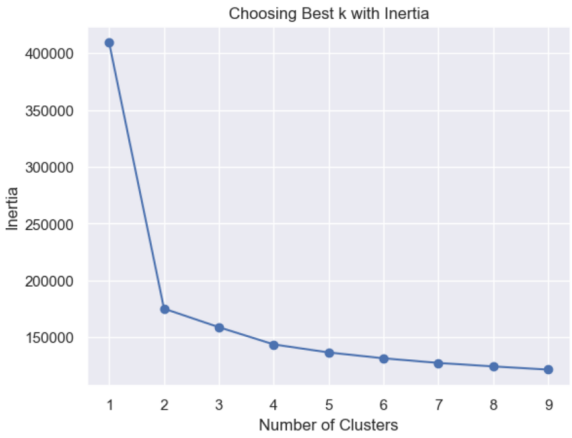


Figure 3: Elbow method for finding best k value

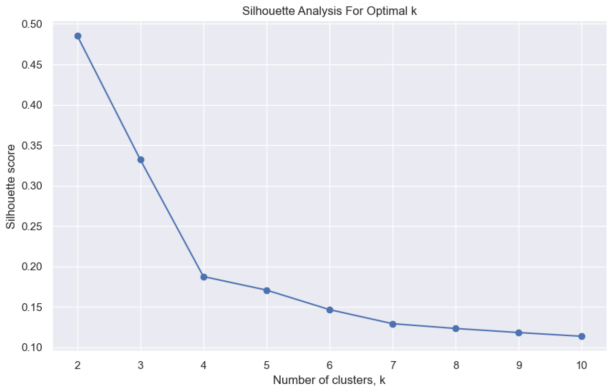
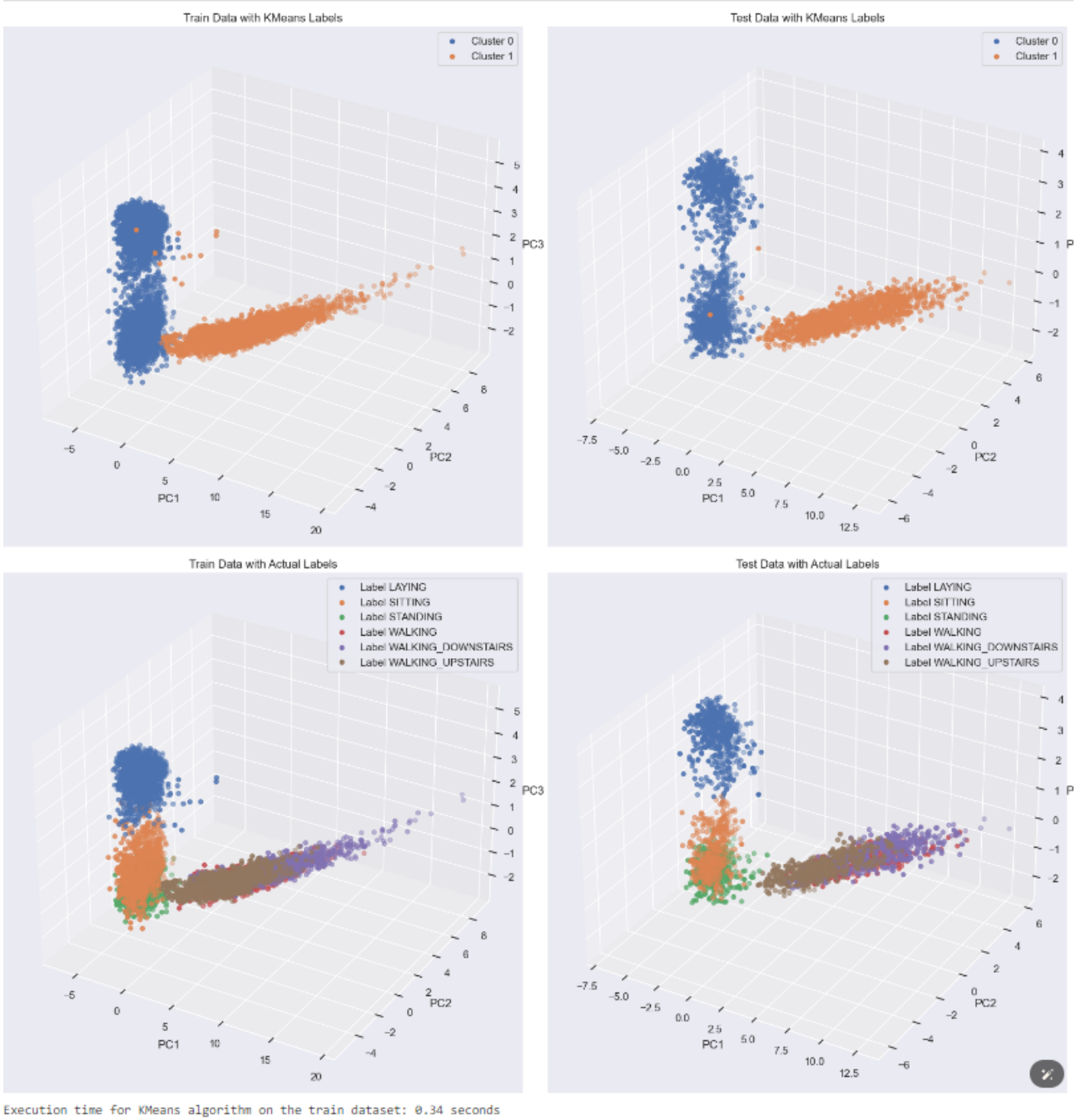


Figure 4: Silhouette method for finding best k value

For the elbow method, the best k value is found by determining the value of k for which the curve starts to decrease gradually. From Figure 3, the value is found to be 2. For the Silhouette method, the best k value is found by determining the value of k for which the silhouette score is highest. The value was found to be 2 as well.

The clustering was then performed for KMeans algorithm by setting the value of k (number of clusters) to 2 and fitting the KMeans model on the training set. The fitted model was then used to predict labels for the train and test sets to ascertain how well the model performs. The obtained results is as illustrated in the figure below.




```
Execution time for KMeans algorithm on the train dataset: 0.34 seconds
```

Figure 5: KMeans clustering on the original dataset

2. **DBSCAN Clustering:** The DBSCAN algorithm requires two hyperparameters to be specified at initialization time. These parameters are the epsilon (which computes the distance between two samples to include them in the same cluster) and 'min samples' parameter which specifies the minimum number of samples a cluster must have. To find the best values for these parameters, an iterative method was employed by computing the silhouette score in each iteration and subsequently choosing the set of values which had the best score. Below is an illustration of how the eps (epsilon) parameter was found.

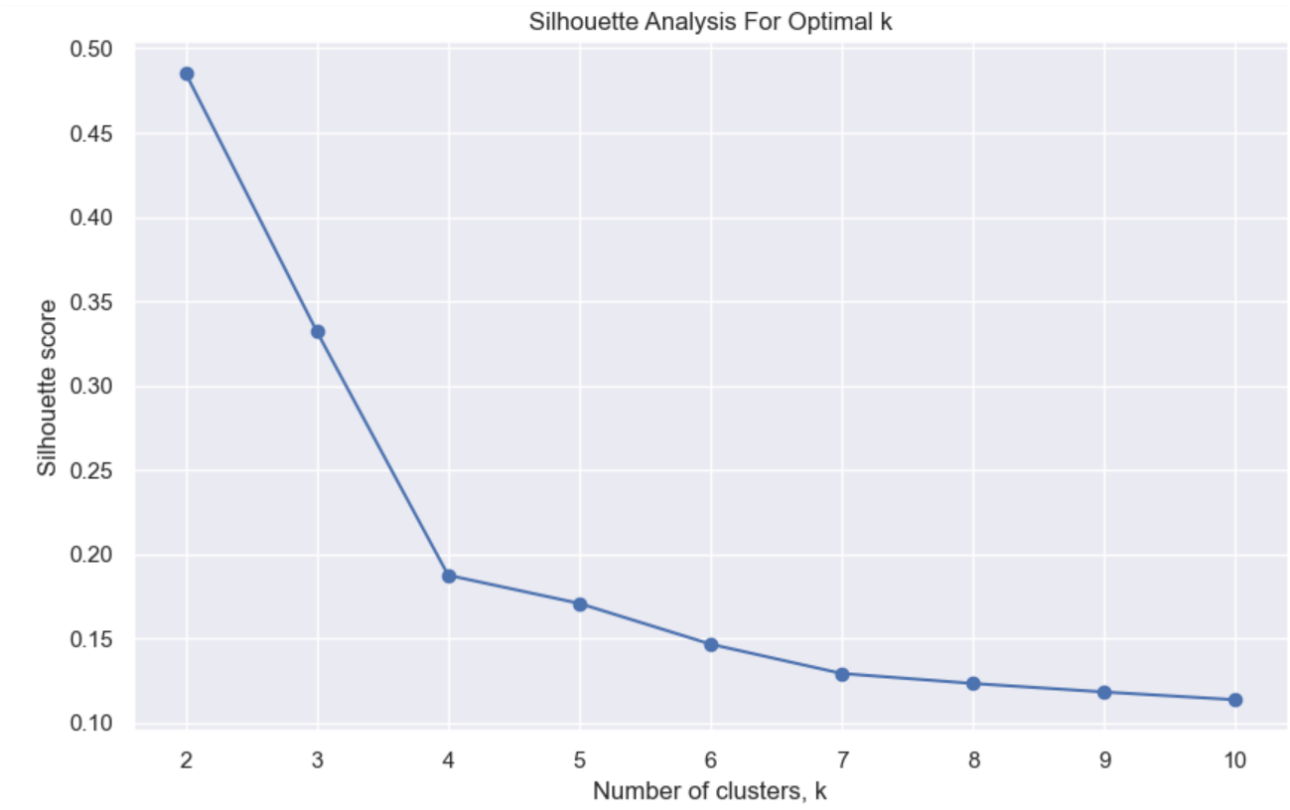


Figure 6: Finding the best value for the eps hyperparameter

With this set of found best parameters, the DBSCAN algorithm was run to determine the clusters found. The image in the figure below illustrates the results from the DBSCAN algorithm.

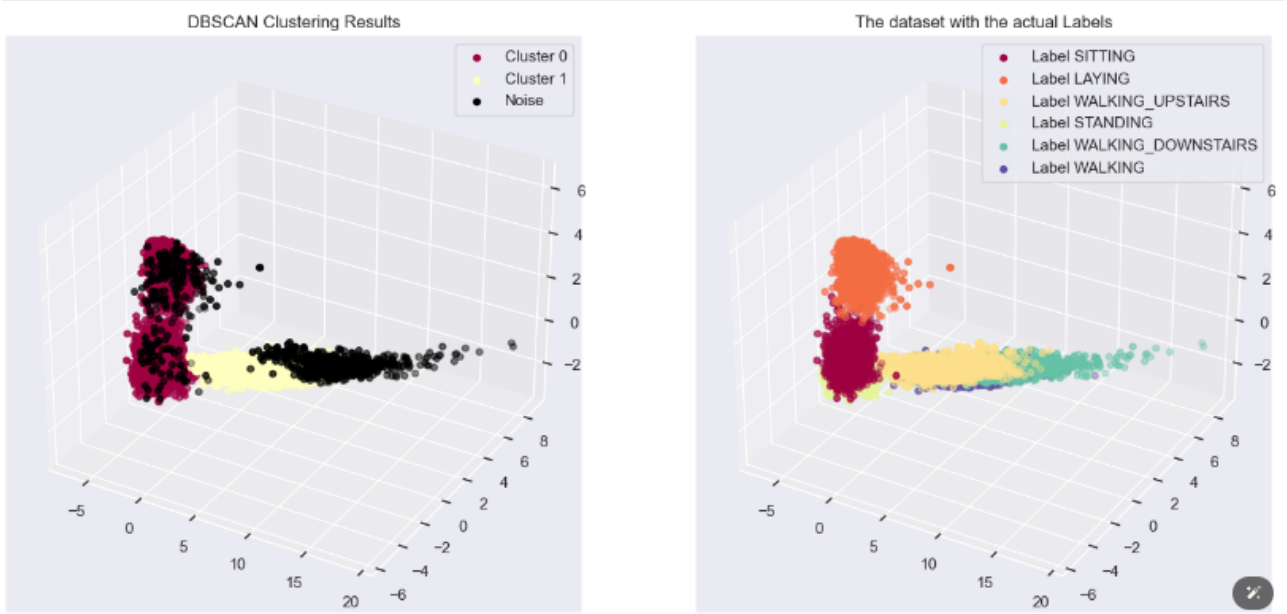


Figure 7: Clusters found by DBSCAN on the original dataset

2.3 Dimensionality reduction Using PCA

For many real word problems, the size of the dataset is often huge that computational time increases considerably. In such scenarios, one might consider an effective way to reduce the computational time. One of the the techniques often employed is dimensionality reduction. Dimensionality reduction techniques work applying certain mathematical formulas such as linear transformations to essentially reduce the dimension of the dataset. One of such techniques is PCA, which reduces the dimension of the dataset by applying linear transformations to combine some features together. The details of how PCA works in principle is outside the scope of this work and can be checked on the internet.

Reasons for choosing PCA for performing the dimensionality reduction:

1. Simplicity: PCA is conceptually straightforward and relies on linear algebra operations that are well-understood and efficiently implemented in most data analysis libraries.

2. Interpretability: PCA transforms the original features into orthogonal components that explain the variance in the data. These principal components can be examined to understand the directions of maximum variance in the data, which can provide insights into the underlying structure of the dataset.

3. PCA creates orthogonal components, which means the resulting features are uncorrelated. This is particularly useful in linear models where multicollinearity can be a problem.

It was found that PCA had essentially reduced the dimension of the original dataset from 561 to 36! Following this, the KMeans and DBSCAN algorithms were re-run for the reduced dataset and the obtained results are as illustrated in the figures below.

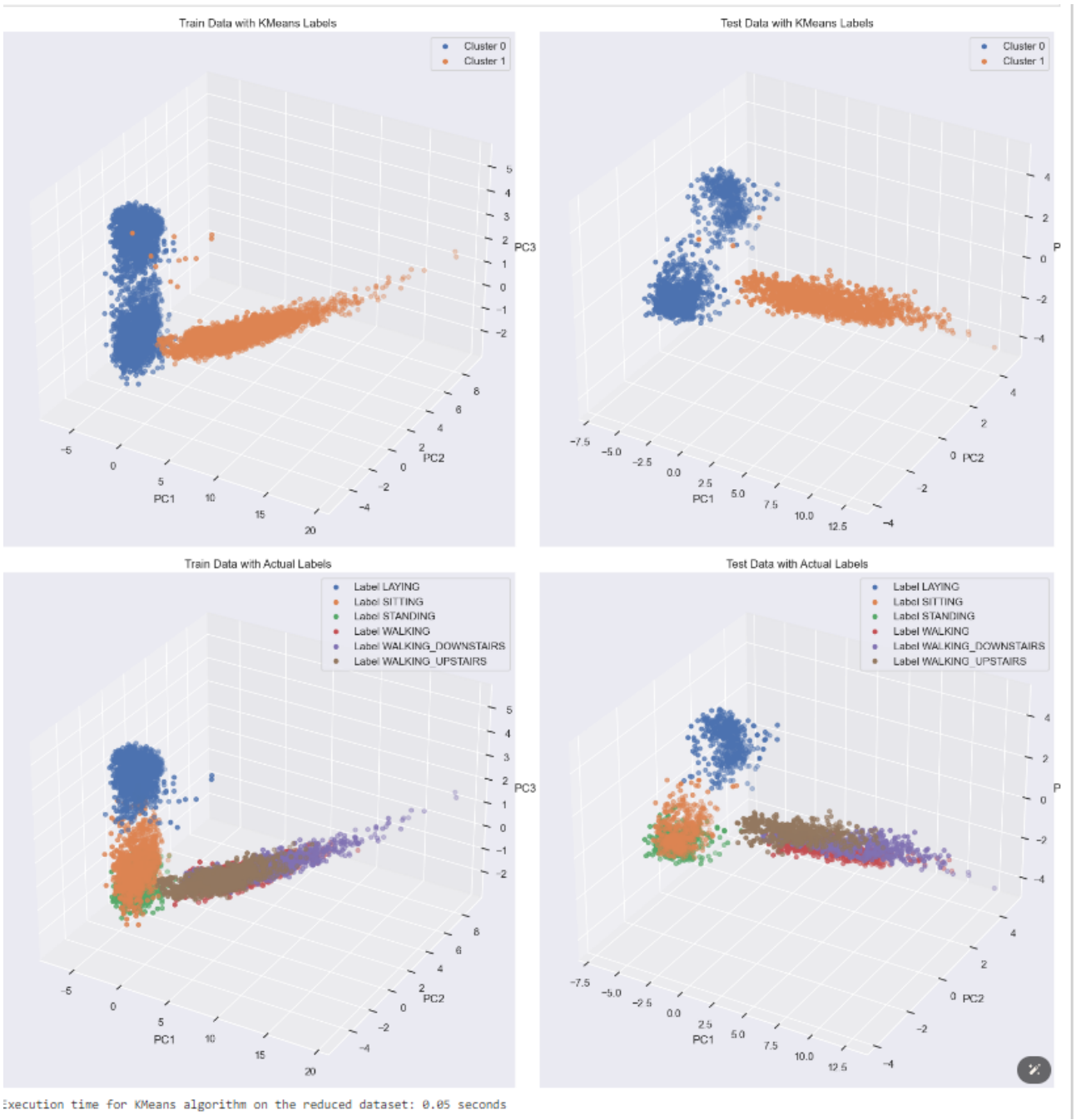


Figure 8: KMeans clustering on the reduced dataset

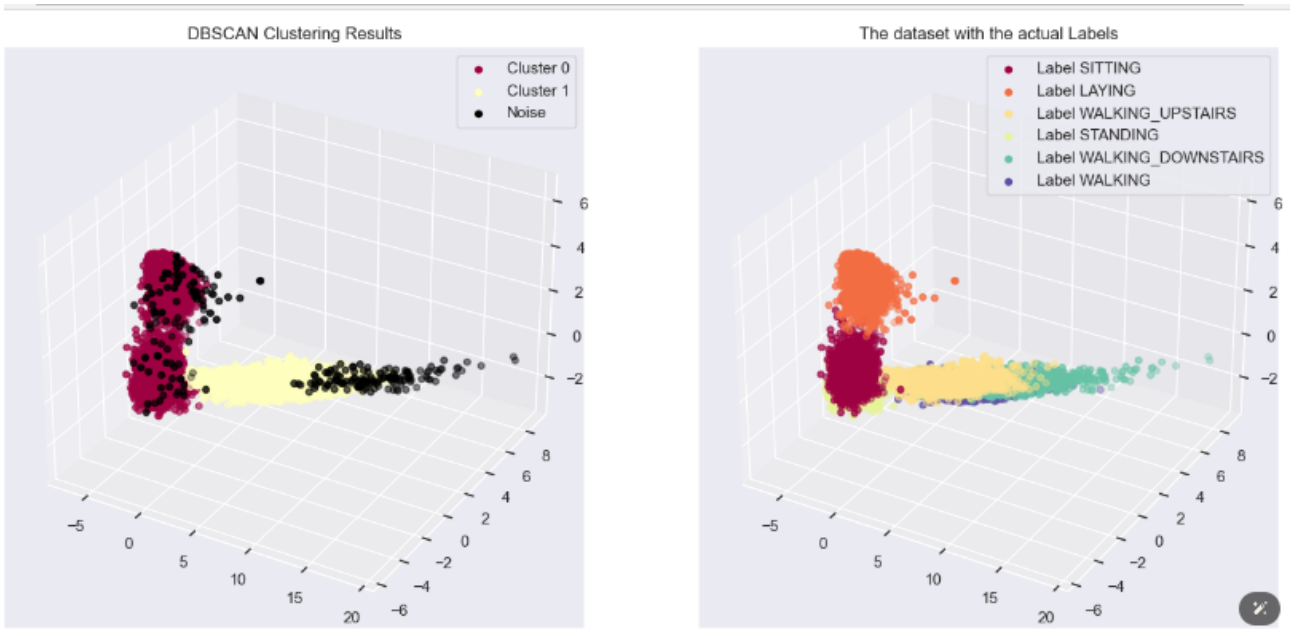


Figure 9: DBSCAN clustering on the reduced dataset

3 Results and Analysis

We can draw several conclusions from the results obtained above. In the following subsections, I discuses some of the paramount results obtained.

3.1 KMeans vs DBSCAN

The results obtained by the KMeans and DBSCAN algorithms on both the original dataset and the reduced dataset show that the KMeans algorithm tries to classify every single data point into one of the available clusters. In doing so, it ends up classifying noise as well. In contrast, DBSCAN found the same set of clusters as that of KMeans, but avoids classifying noise samples. By visualizing the results on 3D plots, the clusters found by DBCAN seem to be more representative of the dataset than that of KMeans, since KMeans classifies noise as well. In addition to this, it appears that DBSCAN classifies the all data points that belong to the label "WALKING DOWNSTAIRS" as noise. The reason could be that the body orientation during this activity is quite different from performing other non-stationary activities such as "WALKING". It also classifies some of the labels belonging to the label "WALING UPSTAIRS" as noise. When we think of the body orientation during the mentioned activities, it becomes clear why DBSCAN tries to classify those labels as noise. Thinking of what the labels mean in real-world scenarios, it becomes obvious that DBSCAN forms a more realistic clusters.

On the other hand, KMeans is computationally more efficient than DBSCAN in terms of the time taken to fit the data.

3.2 Effect of Using PCA

By employing PCA, the dimension of the original dataset reduced from 561 to 36. The effect of this is that it resulted in a better clustering when DBSCAN was used. This claim is backed up by the fact that, on the original dataset, the number of samples classified as noise was 892, while on the reduced dataset, this number significantly dropped to 244. The silhouette score for also increased from about 3.7 to about 4.7 when PCA was applied, hence showing that PCA improved the performance of the DBSCAN algorithm.

Another benefit of using PCA and perhaps, the most beneficial point is that it helped reduce the computational time of both clustering algorithms. The execution time of KMeans algorithm decreased by a factor of approximately 7, while that of DBSCAN dropped by a factor of about 5. This means, employing PCA can significantly reduce computational time (especially when the dataset is of high dimensionality)

	Execution time (seconds)
Kmeans_execution_time_original_dataset	0.334834
Kmeans_execution_time_reduced_dataset	0.042888
DBSCAN_original_execution_time_dataset	1.368936
DBSCAN_reduced_execution_time_dataset	0.255754

Figure 10: Execution time of KMeans and DBSCAN on the original and reduced datasets

4 Conclusion and Reflections

Throughout this project, I have explored how smartphones can be used to understand people's physical activities through the data collected from their built-in sensors. My efforts demonstrated the incredible potential of smartphones beyond communication, highlighting their capability as tools for analyzing and interpreting human movements.

Several challenges were faced, such as transforming the dataset into one single file, and finding the best hyperparameters for the clustering algorithms employed in this project. Despite these hurdles, the findings made have exciting implications. The two algorithms used, both found two clusters which suggests that the dataset can mainly be distinguished by stationary and non-stationary activities.

In conclusion, by making use of clustering algorithms such as KMeans and DBSCAN, it is possible to classify human activities into clusters which can then be used for further analysis. By employing dimensionality reduction techniques, one can essentially enhance the computational efficiency of these algorithms.