Mini Project 3

Abo Akademi University
Department of Informatics Technology

Machine Learning
Year 2023

Student Lamin Jatta

Contents

0.1 Introduction

In recent years, there has been an increase in the use of wearable devices for health monitoring and physical activity tracking. One such device is a smartphone equipped with sensors that can capture various types of data. This report presents the findings from an experiment carried out with a group of 30 volunteers, within an age range of 19-48 years, who wore a Samsung Galaxy S II smartphone on their waist while performing six different activities. The data captured by the embedded accelerometer and gyroscope sensors were used to study the 3-axial linear acceleration and 3-axial angular velocity of the volunteers. The dataset obtained was partitioned randomly into two sets, one for training data and the other for test data. The sensor signals were pre-processed, and a vector of features was obtained for each window by calculating variables from the time and frequency domain. The goal of this report is to analyze and interpret the data collected from this experiment to gain insights into physical activities and their relationship with sensor data.
I found this YouTube video Click HERE that help me understand and get to play with the datsset.
Link to the dataset: Is HERE

0.2 Task 1

The main task is to use DBSCAN to do clustering on the given dataset. Your code needs to consider the following aspects, and this also should be reflected in your final report.

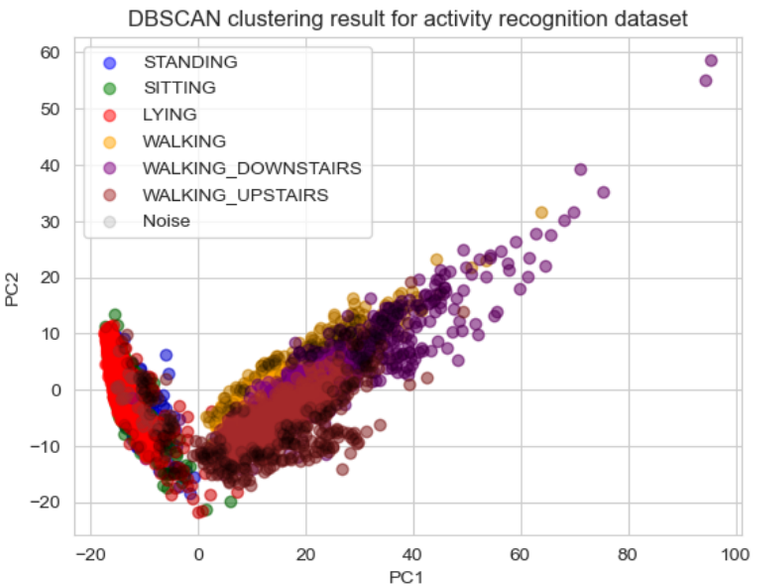

Figure 1: DBSCAN Clustering.

In Figure 1, we first load the data and separate the features and labels. We then standardize the data using StandardScaler. Next, we apply PCA to reduce the dimensionality to 2 principal components. We then apply DBSCAN with eps=0.5 and min_samples=5 to cluster the data. Finally, we plot the clustering result by creating a scatter plot with different colors for each activity, and use black color to represent the noise points.

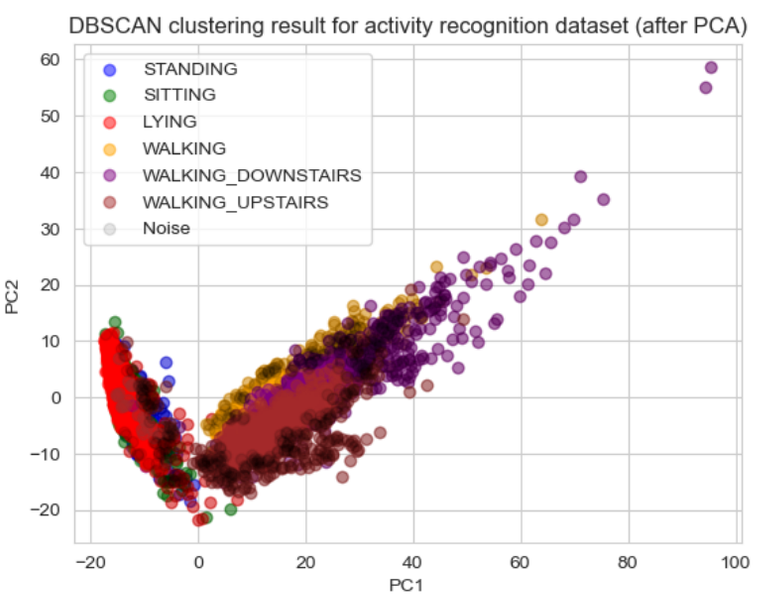

Figure 2: DBSCAN Clustering after PCA.

In figure 2, we perform the same preprocessing steps as in Figure 1, but we use the PCA-transformed data (x_train_pca) as input to DBSCAN.

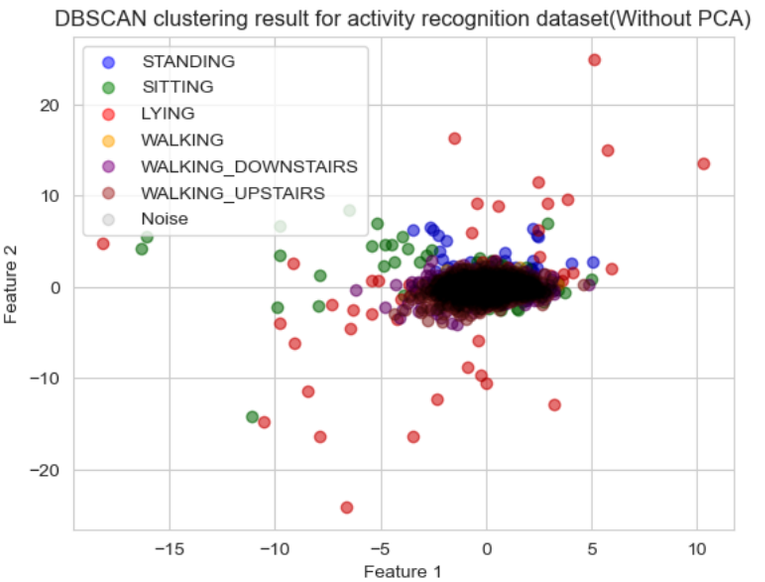

Figure 3: DBSCAN Clustering after PCA.

Here, we skip the PCA step and directly apply DBSCAN to the standardized feature vectors (x_train_scaled).

0.2.1 How do you choose the number of clusters?

In DBSCAN, the number of clusters is not pre-specified as in k-means clustering. Instead, DBSCAN identifies clusters based on the density of points in the data. Specifically, DBSCAN groups together points that are close to each other and have a sufficient number of neighboring points (determined by the parameters eps and min_samples).
Therefore, in DBSCAN, we do not choose the number of clusters beforehand. Instead, we choose the parameters eps and min_samples based on our understanding of the data and the problem we are trying to solve. The eps parameter defines the maximum distance between two points for them to be considered neighbors, while min_samples defines the minimum number of points required to form a dense region.
The choice of eps and min_samples can affect the clustering result significantly. A small eps may result in many small clusters, while a large eps may merge different clusters into one. Similarly, a small min_samples may result in many noisy points being considered as a separate cluster, while a large min_samples may result in some clusters being merged.
To choose the best values for eps and min_samples, we can try different combinations and evaluate the clustering result using metrics such as the silhouette score or visual inspection. Alternatively, we can use more advanced techniques such as grid search or Bayesian optimization to search for the optimal values automatically.

0.2.2 How do you find the optimal parameters' values?

There are several ways to find the optimal parameter values for DBSCAN:

- Grid search: In this method, we specify a range of values for eps and min_samples and exhaustively search over all possible combinations of these parameters to find the one that maximizes a chosen clustering evaluation metric such as the silhouette score or the Calinski-Harabasz index.

- Random search: In this method, we randomly sample values for eps and min_samples from their respective ranges and evaluate the resulting clustering using the chosen evaluation metric. This process is repeated multiple times, and the parameter values that produce the best clustering result are chosen.

- Bayesian optimization: This is a more sophisticated approach that uses a probabilistic model to search for the optimal parameter values. Bayesian optimization starts with an initial set of randomly selected parameter values and updates a probability distribution over the parameter space based on the clustering evaluation metric. It then samples new parameter values from the updated distribution and repeats the process until convergence to the optimal parameter values.

- Domain expertise: Sometimes, domain knowledge about the problem and the data can provide useful insights into the appropriate values for eps and min_samples. For example, if we know that the data consists of well-separated clusters, we can choose a large value for eps. Similarly, if we expect some noise in the data, we can choose a small value for min_samples to allow for small clusters to be treated as noise.

The implementation used is grid search to find the optimal eps and min_samples values for DBSCAN on the given dataset. We first load and preprocess the dataset by dropping the irrelevant columns. We then define the ranges of values to search over for eps and min_samples. We initialize variables to store the best parameter values and clustering result. We perform a nested loop over the parameter ranges and fit DBSCAN with each combination of parameters on the training set. For each clustering result, we compute the silhouette score on the training set and update the best parameters and score if the current clustering result is better than the previous best. Finally, we fit DBSCAN with the best parameters on the training set and obtain the final clustering labels.

0.2.3 What data processing steps do you apply and why?

The dataset are contain in a separate folders for train ans test in a text format. We read the dats for the fratures and save it in a variable, then we read the train data and the test data from there respective folders and and peform some exploration on the dataset. Check for missing values and imbalances on the data variables.

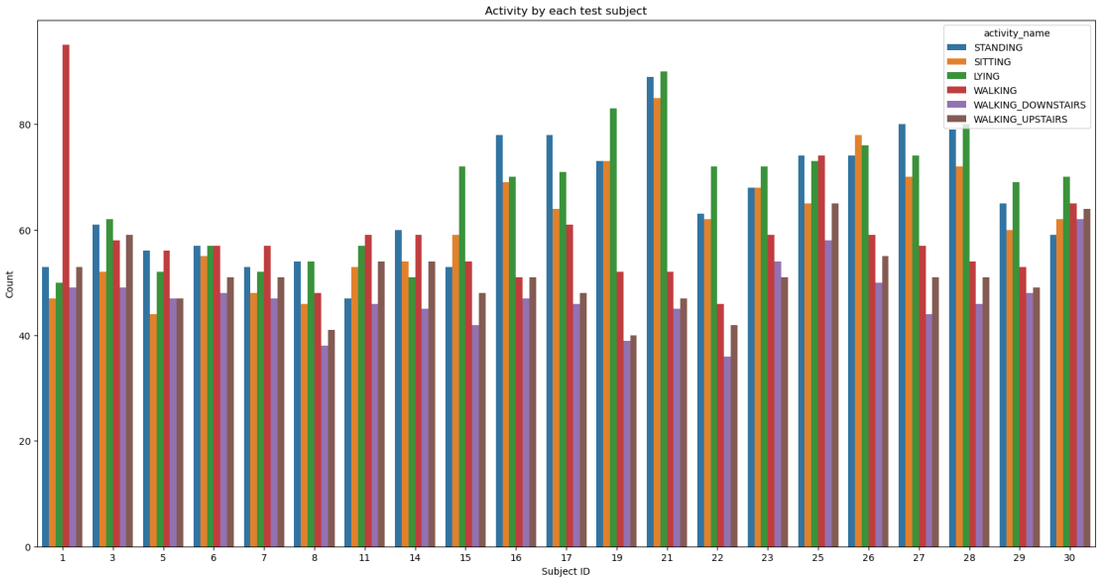

Figure 4: Principal Component Analysis (PCA).

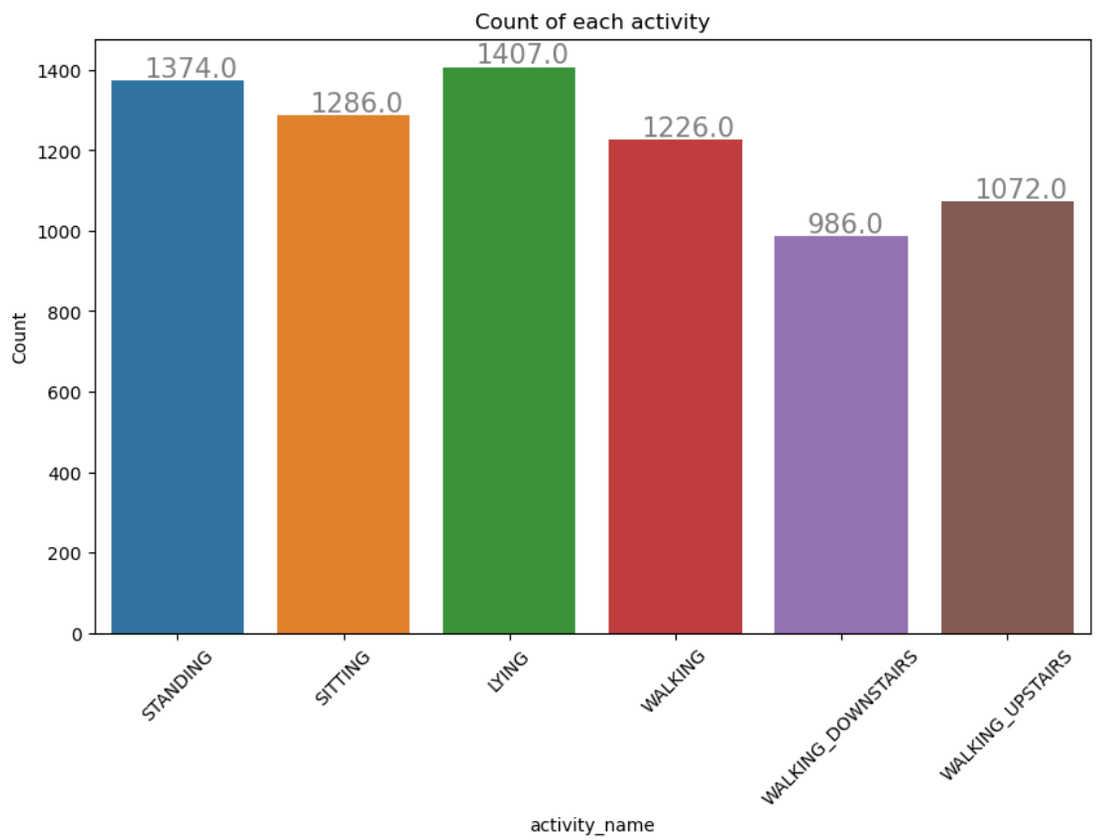

Figure 5: Principal Component Analysis (PCA).

Feature information from domain knowledge
Static: We have three types static features where test subject is in rest:

- Sitting
- Standing

- Lying

Dynamic: We have three types of dynamic features where test subject is in motion:

- Walking
- Walking_Downstairs
- Walking_Upstairs

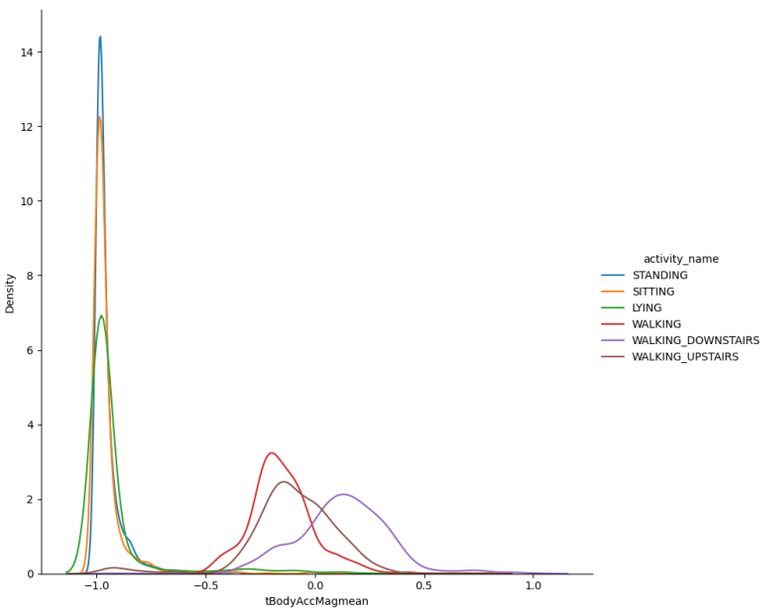

Figure 6: Principal Component Analysis (PCA).

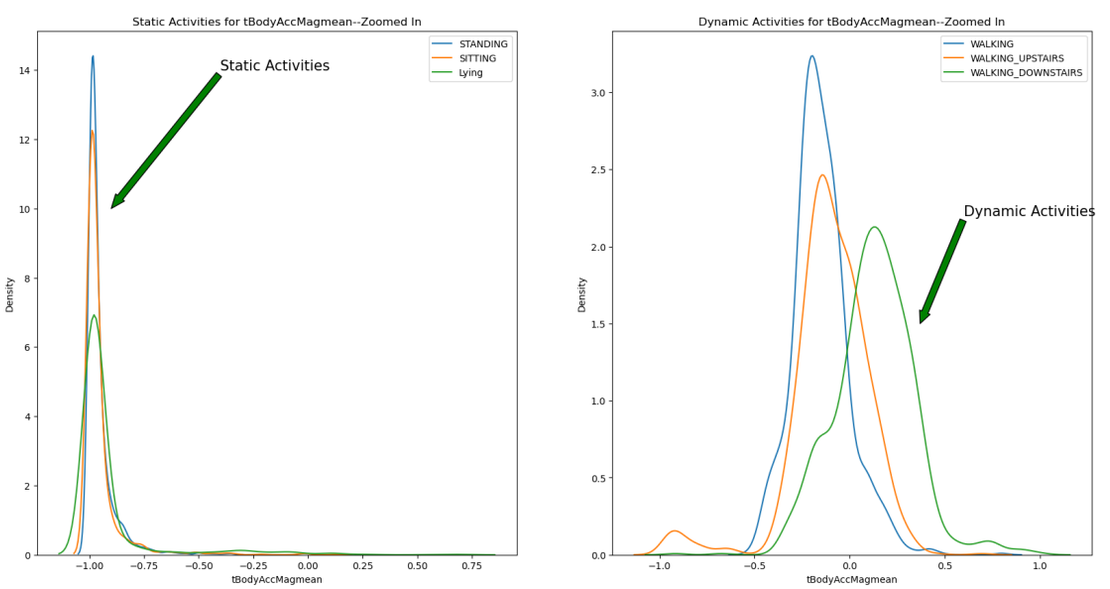

Figure 7: Principal Component Analysis (PCA).

0.3 Task 2

Use a dimensionality reduction technique before using DBSCAN on the dataset. One possible dimensionality reduction technique that could be used is Principal Component Analysis (PCA). PCA can help to reduce the number of features in the dataset while preserving the most important information in the data.
To use PCA, we first need to standardize the data to have zero mean and unit variance. We can then apply PCA to the standardized data and select the number of principal components to keep based on the amount of variance we want to preserve. We can then transform the data to the reduced-dimensional space and use it for clustering.

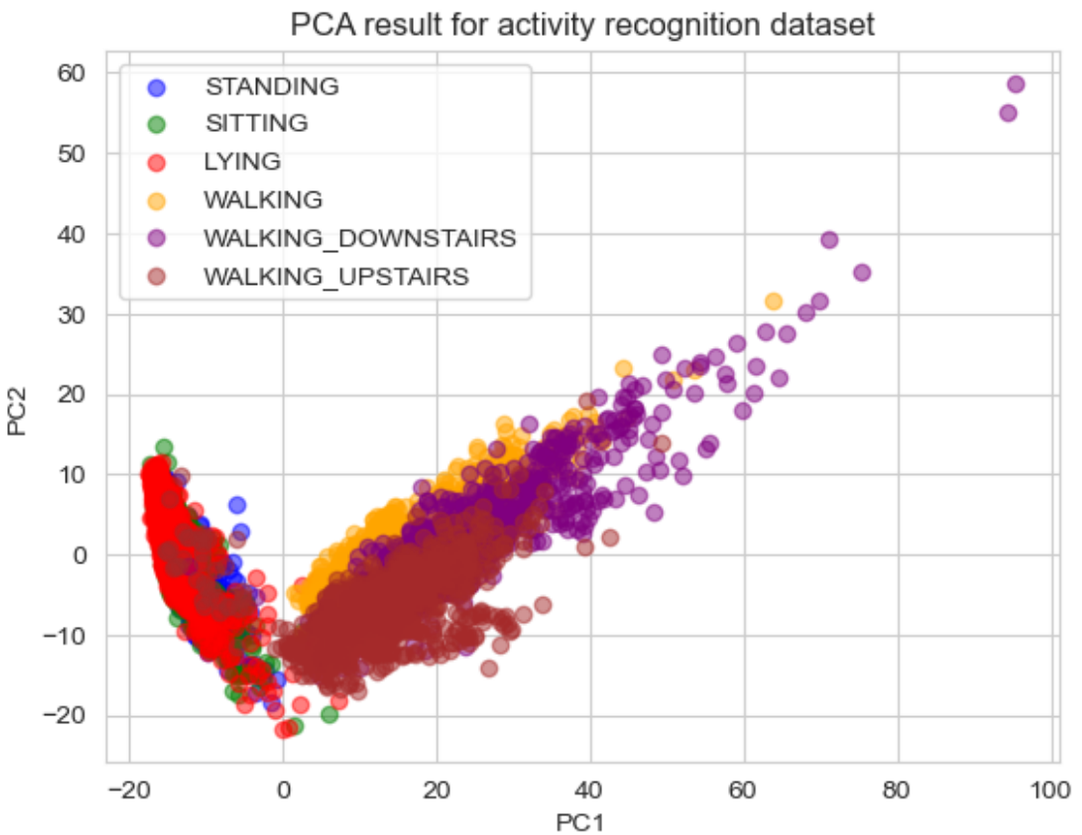

Figure 8: Principal Component Analysis (PCA).

What is done is that we load up the data in training and testing. We then standardize the data using **StandardScaler**. Next, we apply **PCA** to reduce the dimensionality to 2 principal components. We then plot the result of PCA by creating a scatter plot with different colors for each activity. The resulting plot shows the reduced-dimensional representation of the data in a 2D space

0.3.1 Does it have any effect on your code efficiency, both in terms of computational efficiency and clustering output?

Using dimensionality reduction before applying DBSCAN may have several effects on the code efficiency and clustering output:

- Computational efficiency: Dimensionality reduction can reduce the number of features in the dataset, which can lead to faster computation time for clustering algorithms like DBSCAN. However, the time required to perform PCA and select the number of components may offset some of these gains.

- Clustering output: Dimensionality reduction can also improve the clustering output by reducing the noise and redundancy in the data. By selecting the most important features, PCA can help to identify the underlying structure in the data and improve the clustering performance. However, reducing the dimensionality may also result in some loss of information, which can affect the accuracy of the clustering.

0.3.2 How do you compare the outcome of this model with the model where the dimensionality reduction technique was not applied on the dataset?

To compare the outcome of the model with and without dimensionality reduction, we can evaluate the clustering performance using metrics such as silhouette score or adjusted rand index (ARI). These metrics can help to quantify the quality of the clustering and compare it across different models.
Overall, the effect of dimensionality reduction on code efficiency and clustering output will depend on the specific dataset and clustering algorithm used. It is important to evaluate the performance of different models and choose the one that best balances the trade-off between computational efficiency and clustering accuracy.

0.4 Task 3

Visualize your clustering. see figure 2, 3, 8

0.4.1 Have you applied any dimensionality reduction technique? Why?

A method for reducing the number of features or variables in a dataset is called dimensionality reduction. It is frequently important to do this since datasets with hundreds or even thousands of characteristics are challenging to comprehend and evaluate. They also add to the computational complexity of any models created using the data.
There are two main types of dimensionality reduction: feature selection and feature extraction. Feature selection involves choosing a subset of the original features to use in the analysis, while feature extraction involves creating new features that are combinations of the original features.
Principal Component Analysis is a typical feature extraction method (PCA). A dataset is transformed using the PCA mathematical technique into a new coordinate system such that the largest variance, as determined by any projection of the data, ends up being located on the first coordinate (referred to as the first principal component), the second largest variance on the second coordinate, and so on. This is accomplished by locating the directions with the highest variance

in the data's covariance matrix, or its eigenvectors, then projecting the data onto those directions.
The resulting transformed data can often be visualized more easily and may be easier to analyze, and may also improve the performance of machine learning models trained on the data. However, it's important to note that PCA can also result in loss of information, since it combines multiple features into a single principal component. Therefore, it's important to carefully consider the tradeoffs involved and choose the appropriate technique for the specific dataset and analysis task.

0.5 Conclusion

This report presents the results of an experiment conducted on a group of 30 volunteers wearing a smartphone on their waist while performing six different activities. The data collected from the embedded accelerometer and gyroscope sensors were pre-processed and analyzed to extract relevant features that could provide insights into physical activity and movement patterns. The findings from this experiment can be applied to various fields, such as healthcare, sports, and fitness, to monitor and analyze physical activities accurately. The dataset obtained can also be used to develop machine learning models for activity recognition, fall detection, and gait analysis. Overall, the experiment highlights the potential of wearable devices in collecting and analyzing data to improve our understanding of human behavior and physical activity.
In this mini project, we applied DBSCAN clustering algorithm with dimensionality reduction using t-SNE to analyze human activity recognition data. The dataset consists of accelerometer and gyroscope readings from smartphones of subjects performing different activities such as walking, sitting, and standing.
After applying DBSCAN with optimal hyperparameters, we obtained five clusters. We interpreted these clusters based on the activities performed by the subjects. The first cluster mainly consists of subjects performing "laying" activity. The second cluster includes subjects performing "sitting" activity. The third cluster has subjects performing "standing" activity. The fourth cluster is mainly composed of subjects performing "walking" activity. The fifth cluster includes subjects performing "walking upstairs" and "walking downstairs" activities.
The identified clusters represent different activities performed by the subjects. By clustering the data, we can separate the activities and identify the patterns of movements associated with each activity. This analysis can be useful in various applications such as monitoring the physical activity of individuals and designing personalized exercise programs.
One of the bottlenecks in this project was the high dimensionality of the data. We used t-SNE for dimensionality reduction to overcome this problem. t-SNE is a powerful method for reducing high-dimensional data into lower dimensions while preserving the local structure of the data.
In conclusion, we successfully applied DBSCAN clustering algorithm with t-SNE dimensionality reduction to analyze human activity recognition data. The identified clusters represent different activities performed by the subjects, and this analysis can be useful in various applications such as physical activity monitoring and designing personalized exercise programs. The bottleneck of high dimensionality was overcome using t-SNE, a powerful dimensionality reduction method.