Exploring Clustering with DBSCAN: Analysis, Optimization, and Visualization

Submitted By: Saad Waseem | saad.waseem@abo.fi

In the field of data analysis, clustering is a widely used technique to group similar data points together based on certain characteristics. One of the popular clustering algorithms is Density-Based Spatial Clustering of Applications with Noise (DBSCAN). In this project, we will use DBSCAN to cluster a given dataset and explore various aspects of this algorithm. The project is divided into three tasks. The first task focuses on how to choose the number of clusters and how to find the optimal values for *epsilon* and *min_samples* in DBSCAN. The second task explores the impact of PCA – A dimensionality reduction technique before using DBSCAN. We will analyze both the computational efficiency and the quality of the clustering output. Finally, in the third task, we will visualize the clusters obtained from the previous tasks. This will give us insights into the clustering structure and help us to better understand the dataset.

Data Analysis

Before performing DBSCAN clustering on the provided dataset on Human Activity Recognition (HAR). I performed few basic data analysis operations to check for data duplication, null values and if there is any possible bias in the dataset. As shown in figure – 1 there are 6 different types of values about human activity and they are fairly balanced.

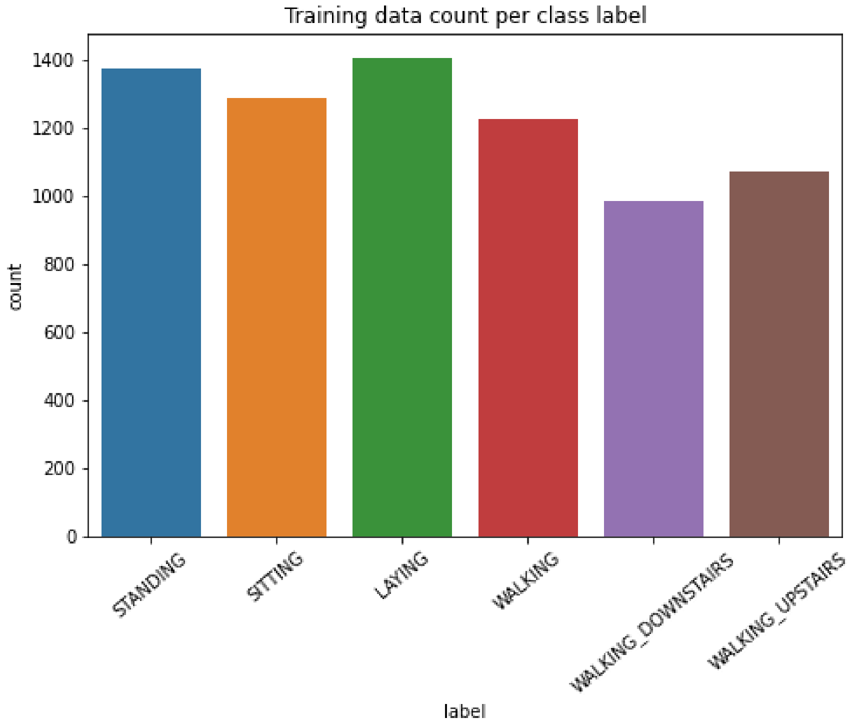


Figure - 1

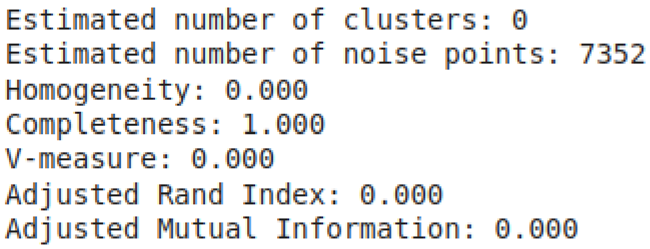
Data Processing

The shared dataset contains balanced distribution of human activity. And the values were already standardized to scale so there was no need of performing additional processing before performing clustering.

DBSCAN Clustering

As DBSCAN is a density-based clustering algorithm that groups together data points that are in high-density regions and separates out those in low-density regions. It uses two important parameters: epsilon (ε) and minimum number of points (MinPts).

When I executed DBSCAN with default parameters, it did not perform optimally for the given dataset and characterized all the data as noise as shown in figure-2. This clear shows selection of these parameters are crucial for DBSCAN to perform well on clustering for the given dataset.



```
Estimated number of clusters: 0
Estimated number of noise points: 7352
Homogeneity: 0.000
Completeness: 1.000
V-measure: 0.000
Adjusted Rand Index: 0.000
Adjusted Mutual Information: 0.000
```

Figure – 2

Parameter Selection

For selecting optimal paramters for DBSCAN clustering I used a standard method known as Elbow method that helps us in choosing the epsilon value by plotting distances for each point and its kth nearest neighbor (k-distance graph) as shown in figure-3 and identifying the "elbow" point in the graph. Accordingly I chose min_samples which is very much dependent on the domain knowledge.

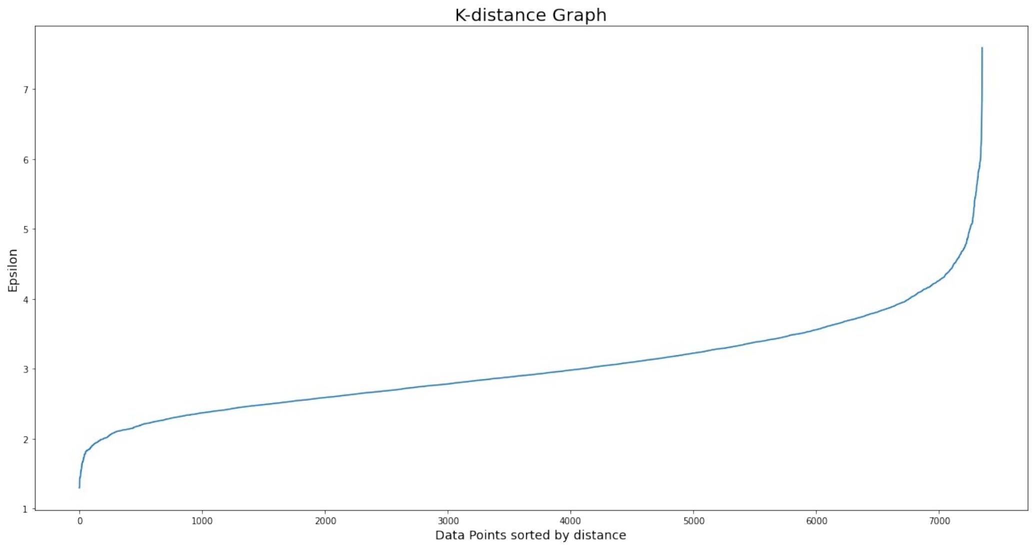
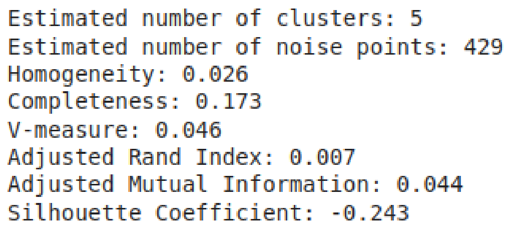


Figure – 3

With epsilon = 4.5 and min_samples = 7 DBSCAN did a nice job by identifying 5 different clusters and rest of the points clustered as noise as shown in figure – 4



```
Estimated number of clusters: 5
Estimated number of noise points: 429
Homogeneity: 0.026
Completeness: 0.173
V-measure: 0.046
Adjusted Rand Index: 0.007
Adjusted Mutual Information: 0.044
Silhouette Coefficient: -0.243
```

Figure – 4

Dimentionality Reduction – PCA

As the provided dataset on HAR is high dimentional data so I applied Principal Component Analysis to reduce more than 500 features in 3 dimentions. This reduction in size helped us make DBSCAN computationally efficient and also improved clustering quality.

As this altered the underlying data for that reason I plotted the K-distance graph with dimensions to calculate optimal values for epsilon and min_samples. After performing PCA, epsilon has changed to 0.3 and min_samples set to 17. This configuration gave us 6 clusters each identifying respective activity as shown in figure – 5

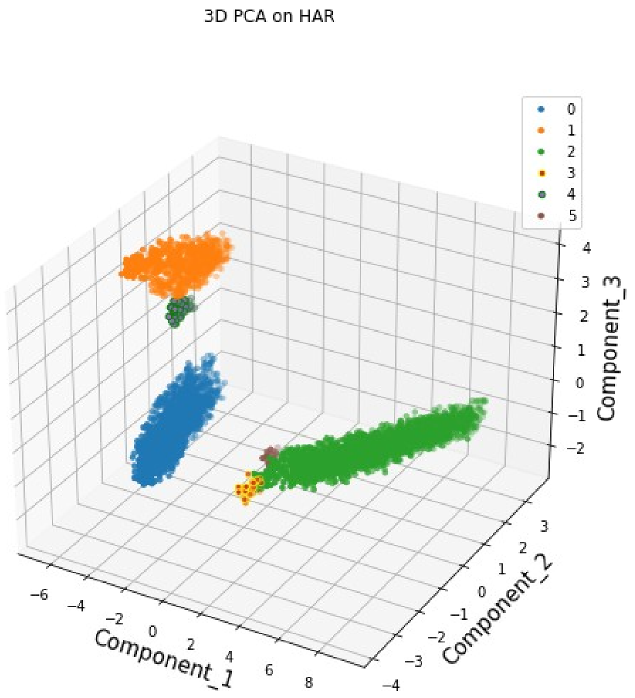


Figure – 5

Comparison with and without PCA:

Without PCA, DBSCAN has to deal with high dimentional data although there were 5 different clusters but silhouette score was not that convincing and much of the points were clustered under one cluster as shown in figure – 6

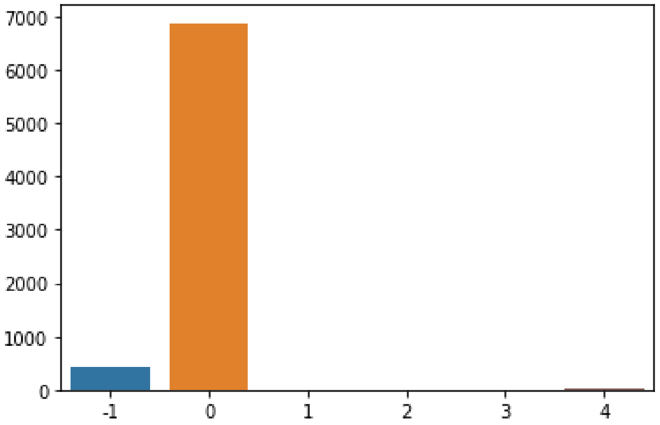


Figure – 6 (Without PCA)

but after applying PCA I was not only able to achieve computational efficiency because now DBSCAN has less data to process but also I was able to get better clustering output. As shown in figure – 7

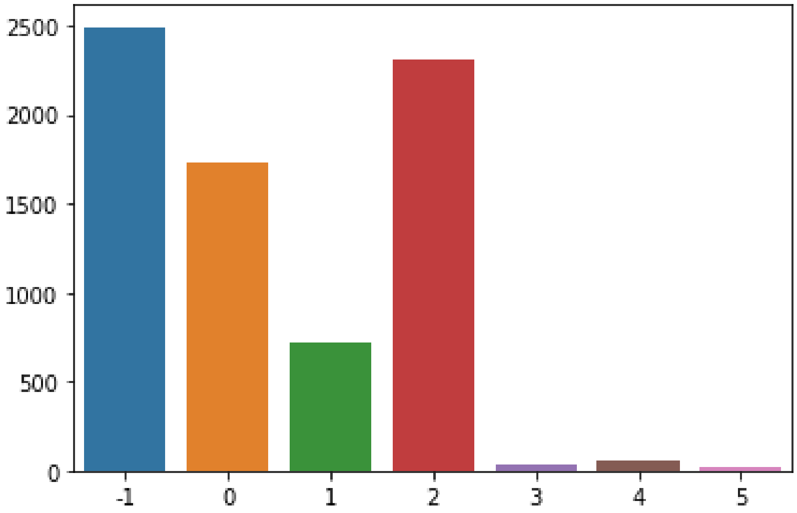
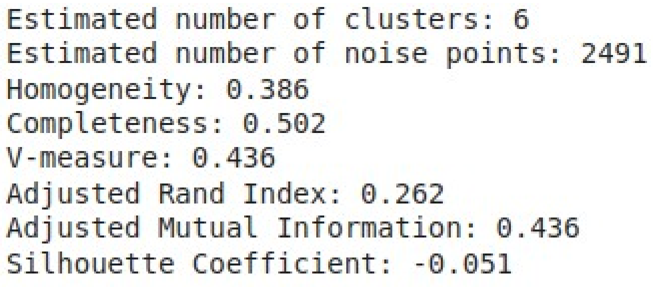


Figure – 7 (With PCA)

Additionally silhouette score also improved with PCA as shown in figure – 4 and figure 8. As the value is closer to 0 in figure – 8 this shows that there are overlapping clusters present in the output. By Fine tuning the clustering algorithm this can be further improved.



```
Estimated number of clusters: 6
Estimated number of noise points: 2491
Homogeneity: 0.386
Completeness: 0.502
V-measure: 0.436
Adjusted Rand Index: 0.262
Adjusted Mutual Information: 0.436
Silhouette Coefficient: -0.051
```

Figure – 8

Conclusion

Clustering is used in many practical applications like recommendation systems and market segmentation. DBSCAN particularly peforms well on data with noise. Choosing the parameters, epsilon and min_samples was crucial to the task as otherwise the algorithm identify everything as a noise. K-Distance graph a.k.a Elbow Method helped in choosing the optimal values for epsilon and min_samples. Eventually to gain computational efficiency, PCA was applied to reduce high dimentional dataset into 3 dimensions and to improve clustering quality. Finally the results were visually reported by plotting a 3D graph. As a future work, the implementation can be extended to relate clusters with actual labels for evaluating clustering accuracy.