Machine Learning 2024 – Mini Project 3

Introduction

The purpose of this project was to analyze a dataset consisting of sensor data produced by a group of volunteers performing certain actions while carrying a smartphone. The actions performed were six distinct activities: walking, walking upstairs, walking downstairs, sitting, standing, and laying. The specifics of these actions in this case are not that important as the goal of the project was to perform clustering on the data to get familiar with clustering techniques and dimensionality reduction.

The dataset consists of 561 columns that have been normalized to a value between -1 and 1. There is also a label for the activity performed, which will not be used in the clustering but may be used for validation of the results.

The project is executed in a python Jupyter notebook, using libraries sklearn for the clustering and dimensionality reduction, and matplotlib for the visualization.

Data processing

First step to be done is loading the data and exploring it to see what issues may need to be addressed before proceeding to the clustering and dimensionality reduction.

Data load

The data was provided in a .txt format, with the columns separated with spaces. The file was opened in the jupyter notebook with the open() function and a pandas DataFrame was constructed from the lines in the text file.

Exploration

Once data was loaded it was explored for irregularities. The dataset is however preprocessed, so there were no irregularities to be found.

Clustering and dimensionality reduction

The clustering was performed with two algorithms, DBSCAN and K-means, both of which are provided by the sklearn library. Dimensionality reduction techniques used were PCA and t-SNE, also from sklearn. The clustering was attempted in multiple stages in different configurations to evaluate the effect of different dimensionality reduction techniques and different parameter configurations for the algorithms.

K-means

K-means clustering is an algorithm that divides the data into a pre-defined number of clusters. Initially the clusters are centered on randomly chosen datapoints, with the rest of the data points assigned to the nearest cluster center. The cluster centers are then recalculated to better approach the center of the assigned datapoints. This process is repeated until the recalculation of the center points no longer moves the centers a significant amount.

The clustering was attempted with varying numbers of clusters at each step, with the range of clusters being between 2 and 16.

The first clustering was done with no dimensionality reduction. Clustering was done for the mentioned range of clusters and the inertia or WCSS (Within-Cluster Sum of Square) was tracked for each cluster count. WCSS is a measure of the distance between the points and the center of their cluster and the distance between the points in the cluster. From the plotting of these values the performance of each cluster count can be evaluated. The elbow method in clustering is a way to determine a suitable number of clusters for the algorithm. This is done by plotting the WCSS over the count of clusters and looking for the point where the lowering of WCSS starts to diminish. In this case that point was at 6 clusters, which agrees with what we already knew about the data.

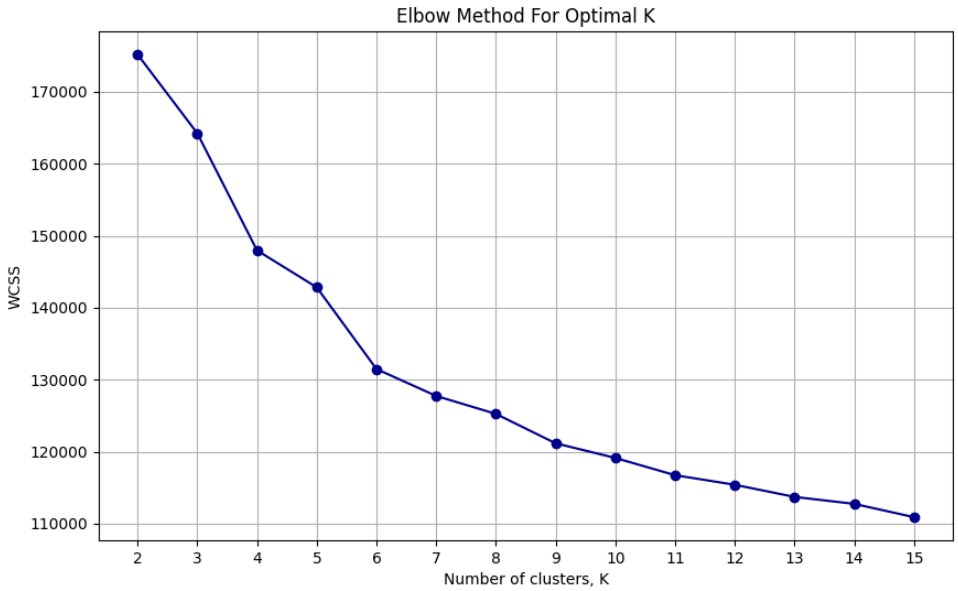

Image 1, Elbow method analysis for K-means clustering, no dimensionality reduction.

Dimensionality reduction was then attempted while still using K-means clustering. PCA, Principal Component Analysis was first tried. PCA calculates the covariance of the features in a linear manner and combines the features to reduce the total number of dimensions. PCA was tested at multiple counts of components (250, 150, 100, 75, 50, 30, 15, 9, 5), and analysis was done similarly as for the non-reduced dataset. Reducing the dimensions did however not necessarily lead to any improved results, and the point where the improvement of WCSS starts to diminish varied with the number of components selected in the PCA.

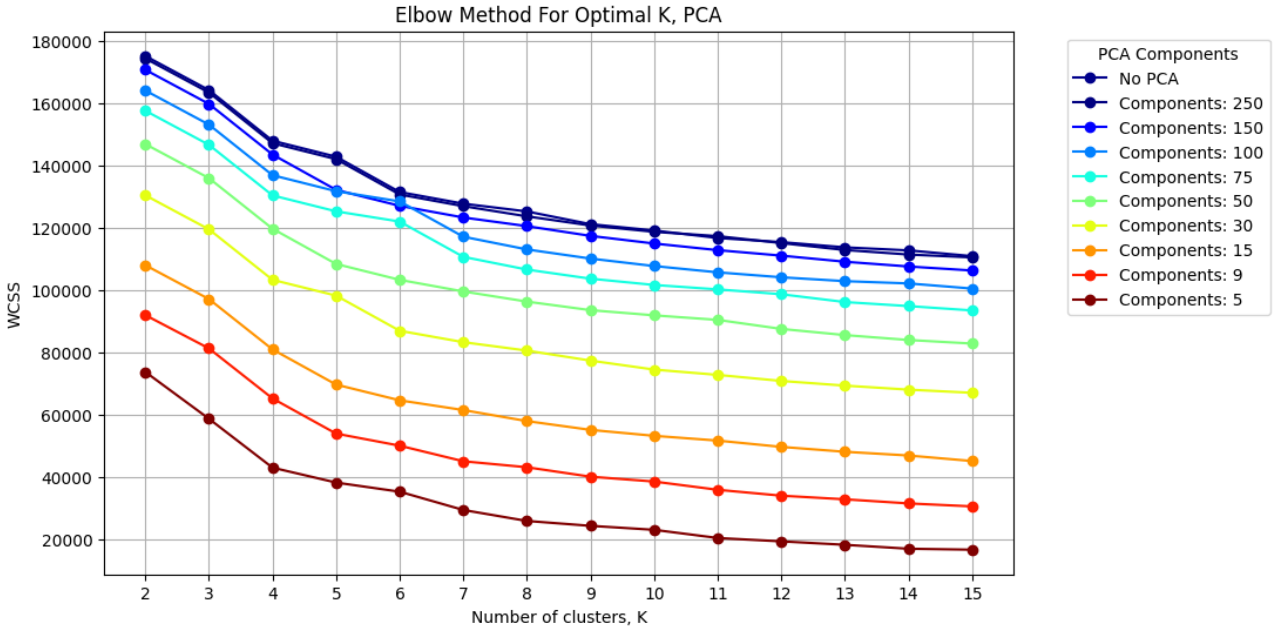

Fig 2, Elbow method analysis over a range of components selected by PCA.

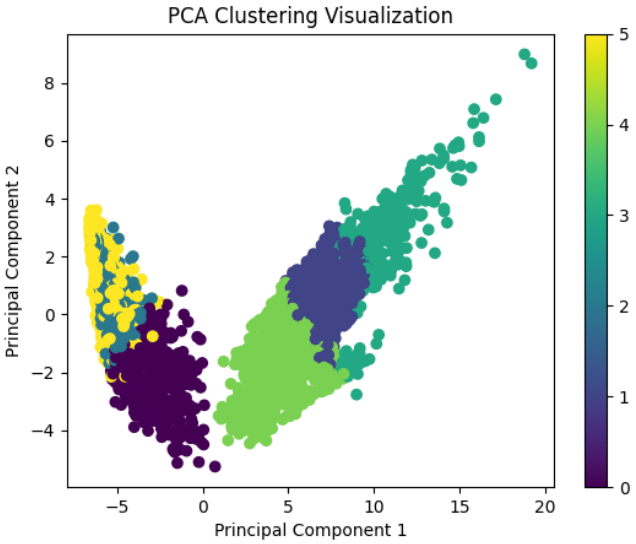

Fig 3, K-means displayed in PCA reduced 2D grid.

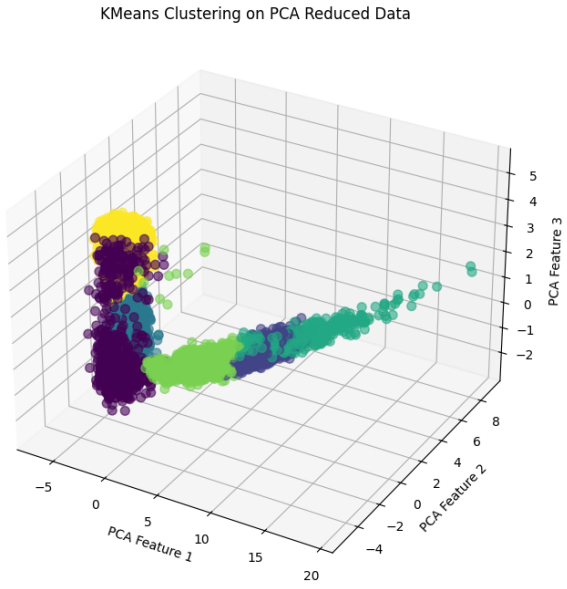

Fig 4, K-means displayed in PCA reduced 3D grid.

Finally, K-means clustering was run with data reduced by t-SNE. t-SNE is a method of non-linear dimensionality reduction that is excellent at preserving information when reducing large numbers of dimensions down to two or three dimensions. Opposed to PCA, t-SNE can pick up on non-linear relationships between features and does better in capturing the relationships between nearby points, while possibly losing some of the overall structure of the data. It is also the more computationally intensive method of the two. The number of dimensions was reduced to three with t-SNE and the same range of clusters were tried. Elbow method analysis was a bit unclear in this case as the reduction in WCSS happened in a smoother curve than for the non-reduced data or for PCA. Visualizing the data clusters in 2D and 3D

however gave an idea of well-formed clusters, and it could definitely be considered that t-SNE outperformed PCA in terms of highlighting the variance in the data.

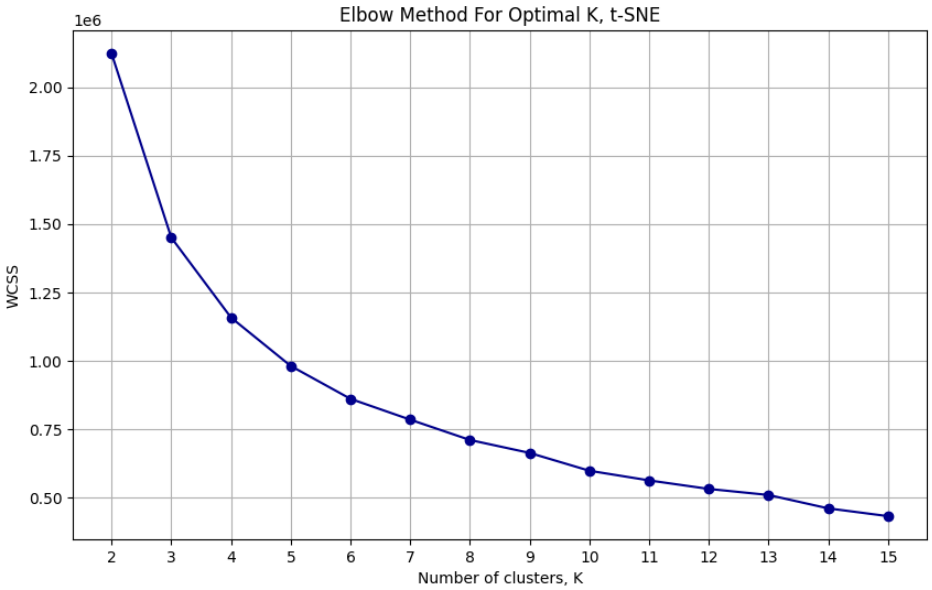

Fig 5, Elbow method analysis for K-means clustering, t-SNE reduction.

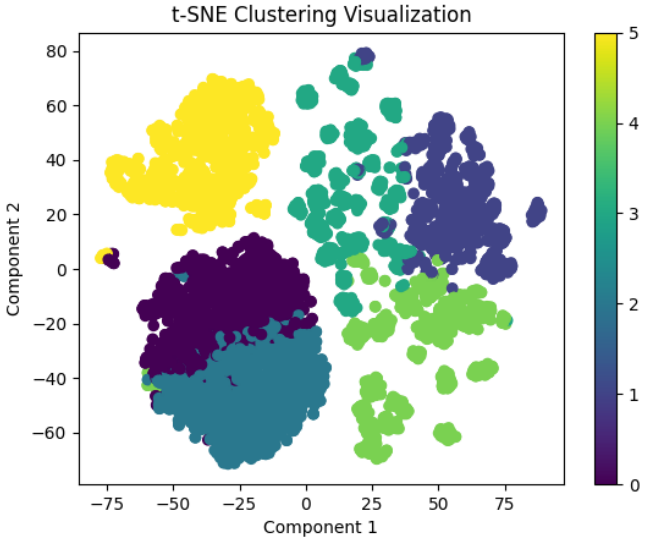

Fig. 6, K-means displayed in t-SNE reduced 2D grid.

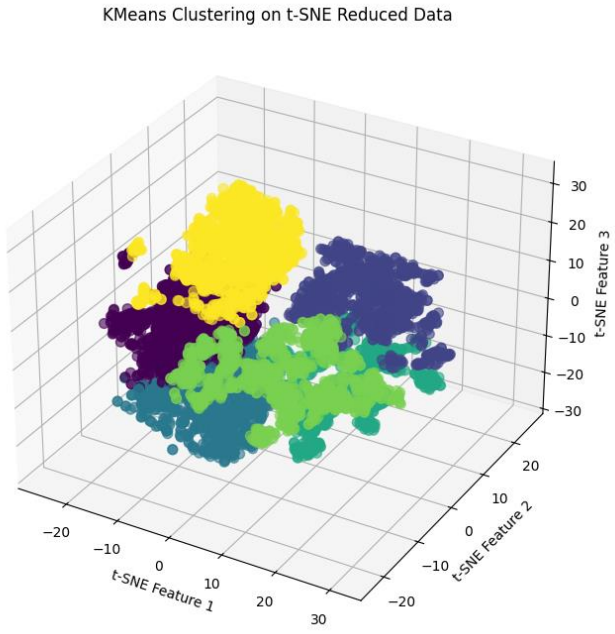

Fig. 7, K-means displayed in t-SNE reduced 3D grid.

DBSCAN

DBSCAN is a clustering method that is instead focused of the density of the data points rather that the distance to the cluster center. There are two parameters to consider for DBSCAN, epsilon or eps, and min_samples. The number of clusters is not defined beforehand but is determined by these parameters. A cluster center is defined when a number of data points *min_samples* can be found within a certain distance *eps*. The difficulty in using DBSCAN comes from the task of finding suitable values for these parameters. One method of finding a good eps value is by using the Nearest Neighbors clustering algorithm to calculate the distance between a number of points nearest to each point in the dataset. A good eps value should then be able to be found where the distance starts to sharply increase. The ideal value appeared to be around 4-6 for a number of min_samples values tested.

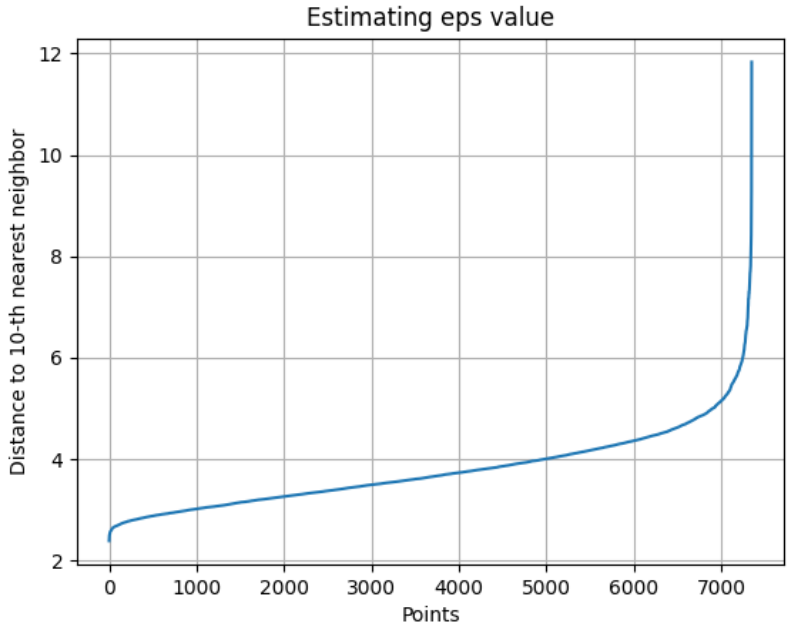

Fig 8, The Nearest Neighbors estimation for the eps value at min_samples=10.

Clustering was then attempted wit DBSCAN without dimensionality reduction. A wide range of parameters were searched for eps and min_samples, however any satisfactory configuration proved hard to find. The number of clusters varied wildly, and a lot of points were classed as noise, meaning that they were not included in any cluster. Additionally, the Adjusted Rand Score was used to validate the clustering against the original classes after having faced issues with finding the best parameters.

eps	min_samples	n_clusters	n_noise	ars
4.5	9	3	489	0.324894
4.5	10	3	499	0.324852
4.5	8	4	463	0.324659
4.5	12	3	538	0.324479
4.5	11	3	523	0.324464
4.5	13	4	555	0.324134
4.5	14	3	579	0.323727
4.5	15	3	599	0.32344
4	11	10	1486	0.313544
4	9	16	1379	0.312432

Table 1, Top results of the DBSCAN clustering without reduction.

Having not achieved great results with the non-reduced data, PCA was performed. Similar as for K-means a range of component counts were tried, and the same ranges of eps and min_samples were scanned. Some changes compared to the non-reduced data were noted. The number of clusters tended to stick to lower values and the level of noise was reduced.

eps	min_samples	n_clusters	n_noise	ars	pca
3.6	29	2	167	0.32954	30
3.5	23	2	176	0.329437	30
3.5	25	2	187	0.329092	30
3.9	3	4	60	0.329012	50
3.8	3	5	69	0.328887	50
3.5	27	2	196	0.328827	30
3.5	29	2	199	0.328786	30
3.9	5	4	85	0.328549	50
4	19	2	175	0.328446	50
3.9	7	3	110	0.328368	50

Table 2, Top performing configurations for DBSCAN with PCA reduction.

Finally for DBSCAN as well, dimensionality reduction with t-SNE was done as well, with the same search for parameters as for PCA. This time though the results showed some improvement. The Adjusted Rand Score showed significant improvement for some configurations and noise was kept fairly low. The number of clusters showed a significant increase.

eps	min_samples	n_clusters	n_noise	ars
3.5	27	22	274	0.564423
3.5	29	23	314	0.560639
3.5	19	18	51	0.520185
3.5	21	19	68	0.519675
3.5	23	22	110	0.510308
3.5	25	21	165	0.508312
3.6	29	20	245	0.350368
4.5	27	9	31	0.324274
4.5	29	9	32	0.324229
3.9	5	11	5	0.320046

Table 3, Top performing configurations for DBSCAN with t-SNE reduction.

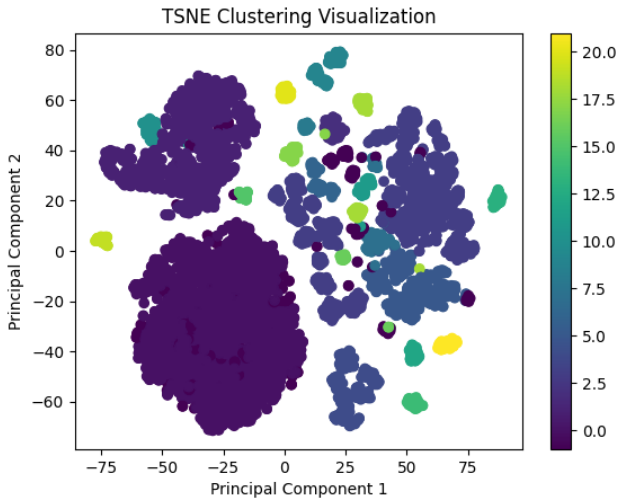

Fig 9, DBSCAN displayed in t-SNE reduced 2D grid.

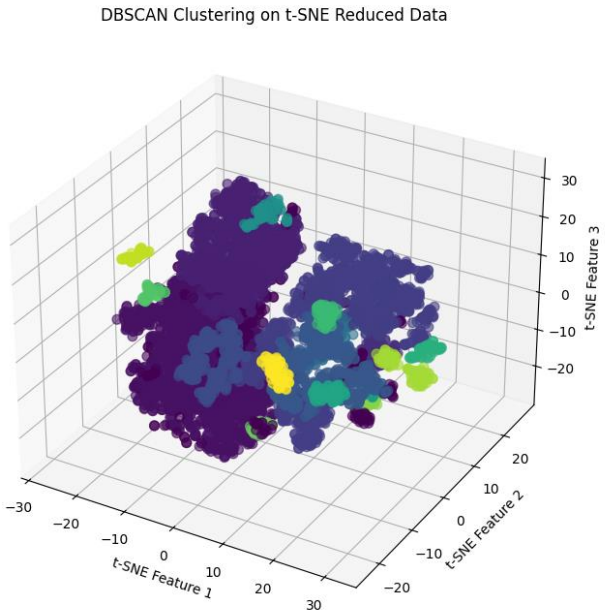

Fig 10, DBSCAN displayed in t-SNE reduced 3D grid.

Conclusions

From reading about it, DBSCAN should be able to perform clustering very well for many different shapes of data. However, for this project DBSCAN did not perform that well, but it may be that due to inexperience the configuration of the parameters was not done correctly and wit the right settings better results could have been reached. To assist in finding the parameters the Absolute Rand Score was used, and this might not be the entirely correct approach to use if this project is to be considered a practice in unsupervised learning as the original labels were then used for validation.

On inspection of the visualizations of PCA versus t-SNE for K-means clustering there is a clear difference in the definitions of different clusters for t-SNE, and based on that you could draw the conclusion that t-SNE is the correct method for reducing dimensionality for this dataset.

References

Pandas Documentation. (2024). Retrieved from https://pandas.pydata.org/docs/index.html

Scikit-learn Documentation. (2024). Retrieved from https://scikit-learn.org/stable/