UPC UB URV

MASTER IN ARTIFICIAL INTELLIGENCE

- Project 2 -

Dimensionality Reduction & Visualization

Introduction to Machine Learning

Professor:
Dr. Maria Salamó
Course: 2022-2023.
Laboratory group 11L

Students:
Lauren Tucker
Mario Rosas Otero

November 13, 2022

1 Introduction

The aim of this project is to analyze the behavior of different PCA implementations and other different dimensionality reduction, clustering, and visualization methods. This report is divided into two main experiments to investigate and visualize the effects of dimensionality reduction. For the first experiment, we wrote our own implementation of PCA and compared it with the PCA and Incremental PCA algorithms from sklearn. For the second experiment, we compared how clustering results are affected by testing different combinations of dimensionality reduction, clustering, and visualization techniques. Through these experiments, we were able to draw conclusions about how each technique manipulates the data.

2 Methodology

2.1 Experiment 1: PCA Comparison & Feature Agglomeration

For this experiment, we wrote our own implementation of PCA (henceforth referred to as OPCA), and we followed the process outlined in the flowchart shown below in Figure 1 to visualize the results and compare them with the sklearn implemenations of PCA and incremental PCA.

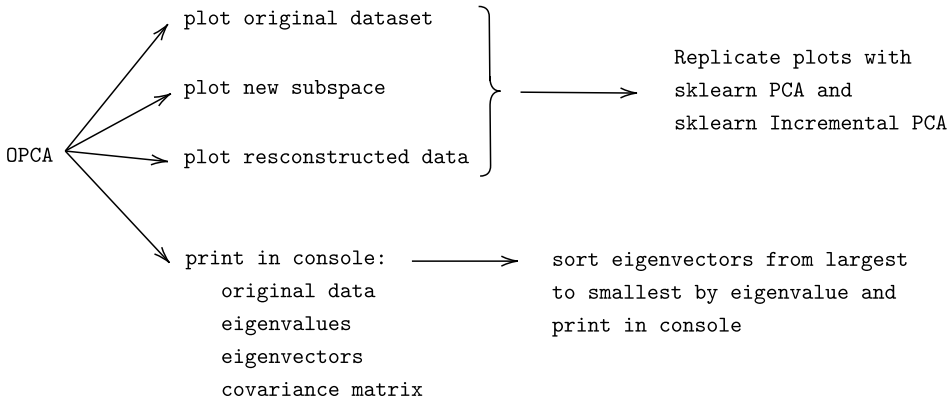


Figure 1: Flowchart of Experiment 1 Process

The OPCA implementation was created from scratch just using some libraries to simplify mathematical operations between data transformation. Specifically, NumPy was used for matrix multiplication and subtraction and SciPy was used to get the eigenvectors from the covariance matrix. The steps followed were mainly as showed during theoretical classes. To enable flexibility of the implementation and allow its use in each required scenario, OPCA is a class with several properties that allow the retrieval of various attributes from it during the experimentation sessions. When the OPCA class is instantiated, both the data and the name of the data must be passed in case visualizations are needed later. The method *fit* within the class makes all the computations needed to transform the data according to the PCA process used and described in algorithm 1, and the only parameter it requires to transform the data is the number of components. When the fit method is computed, the attributes of transformed data, reconstructed data,

eigenvalues, eigenvectors and covariance matrix become available. A *visualize* method was also implemented as part of the OPCA class, which allows the creation of scatter plots of 2, 3 or 4 dimensions from the original, transformed or the reconstructed data automatically and the only strictly required parameter is the labels of the data. Lastly a *scree_plot* method was implemented to output a useful plot of the explained variance ratio of the process, which was really helpful to choose parameters during the posterior experiments.

Algorithm 1: Fit method inside OPCA class

Data: $original\_data$

Result: $transformed\_data$

$data \leftarrow original\_data$; // matrix of samples × features

$axes \leftarrow n\_components$; // number of components to transform data

$means \leftarrow mean(original\_data, 0)$; // mean by features

$sub\_means \leftarrow subtractdata, means$; // subtract the means from each sample

$cov\_mat \leftarrow covariance\_matrix(sub\_means)$; // covariance matrix computation

$eigenvalues, eigenvectors \leftarrow eig(sub\_means)$; // Eigen values and vectors computated

$s\_eigenvalues, s\_eigenvectors \leftarrow sort(eigenvalues, eigenvectors, descending)$; // Sort eigen values & vectors in descending order

$feature\_vector \leftarrow eigenvector[axes]$; // Choosing the important components

$transformed\_data \leftarrow dot\_mult(feature\_vector, sub\_means.T)$; // Dot multiplication of feature_vector and transposed sub_means

$reconstructed\_data \leftarrow dot\_mult(transformed\_data.T, feature\_vector)$; // Dot multiplication of transposed transformed_data and feature_vector

$transformed\_data \leftarrow transformed\_data.T$; // Fixing the transformed_data shape

Implementing the sklearn PCA algorithms was very simple. Like with OPCA, a PCA_sklearn() class was created, which houses functions for performing PCA and Incremental PCA and for visualizing the results. To instantiate a class, we passed the data and its name to the constructor. When calling the functions, the number of components we used corresponded to the total number of components in each preprocessed dataset. As you will see later, this helped us to determine the ideal number of components to use in Experiment 2. After calling the included fit_transform method to transform the data, we were then able to directly retrieve the eigenvalues, eigenvectors, and explained variance ratio from the attributes. We also retrieved the reconstructed data by passing the transformed_data to the inverse_transform method. In the end, these functions return our transformed data, which was used to visualize our results.

Another technique for dimensionality reduction that we will compare against PCA is Feature Agglomeration (FA). For our analysis, we used sklearn's implemenation of FA, following largely the same process as was used for using sklearn's PCA implementations. The primary differences are that we must pass additional parameters to specify the distance metric used and the linkage method, and that rather than retrieving eigenvalues, eigenvectors, and an explained variance ratio, we are retrieving labels.

The visualized results of passing our three datasets through Feature Agglomeration and each of the three discussed implementations of PCA are included in the Results section.

2.2 Experiment 2: Visualization of Clusters with and without Dimensionality Reduction

For this experiment, we followed the process outlined in the flowchart shown below in Figure 2.

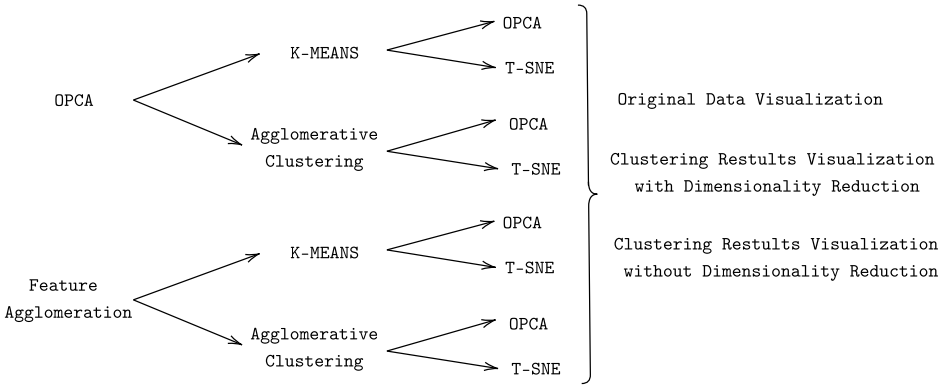


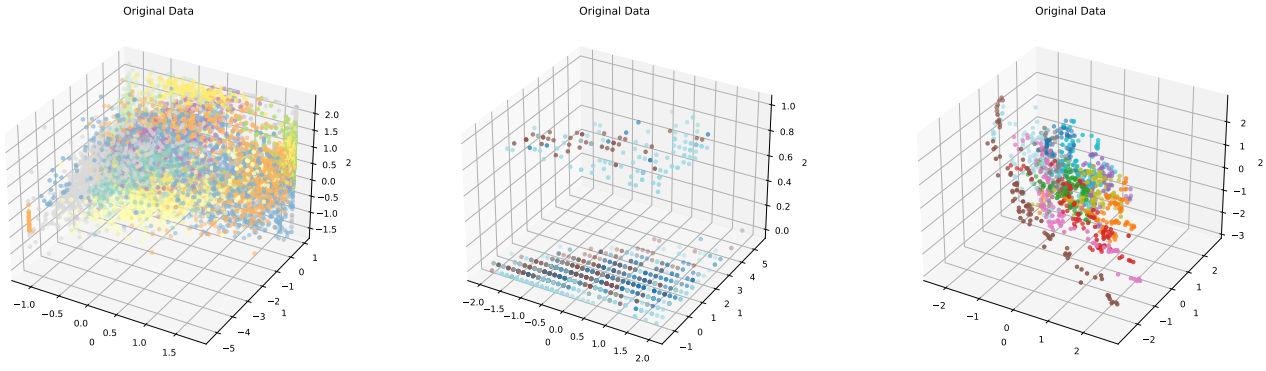
Figure 2: Flowchart of Experiment 2 Process

Using the Pen-based, CMC, and Vowel datasets, we first applied either OPCA, Feature Agglomeration or no dimensionality reduction, then we clustered the resulting transformation using either k-means or Agglomerative Clustering, and then visualized the results using either OPCA or t-SNE.

3 Results

3.1 Experiment 1 Results

First, for reference, we created 3D plots of the first three axes for the original data of each of our three datasets. Since we are only plotting the first three axes, and are only able to view the plot from one point of reference, our interpretations of the visual data are subject to these limitations. The pen-based dataset is numerical, and the CMC and Vowel datasets are mixed, with both numerical and categorical data, and the data was normalized with the same preprocessing function from Work 1. As seen in Figure 3, the shape of each data set is different from the others. The Pen-based data is in a sideways square pyramid shape where the points are generally grouped by class, but with much intermixing. There are also many points stacked around the edges at the base of the "pyramid", likely due to the nature of the preprocessing. The CMC data is in two parallel planes along the z-axis, with no clearly differentiable groups between classes. It seems that some data points are possibly overlapping with each other, since the number of visible data points is much lower than the number of data points in the actual dataset. Finally, the Vowel dataset is roughly ellipse-shaped, with some more clearly identified groupings than the other two datasets, and some classes more sparse and spread out than others, such as the brown class.



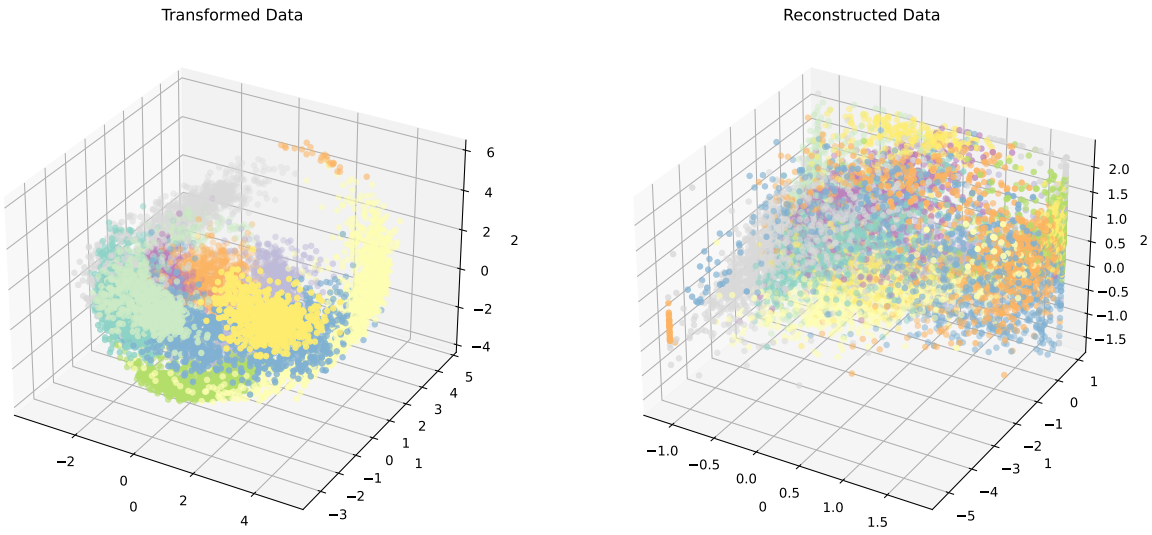
(a) Pen-Based Dataset (b) CMC Dataset (c) Vowel Dataset

Figure 3: Original data projection of Pen-based, CMC & Vowel Datasets

3.1.1 sklearn PCA

In the following plots generated from the sklearn PCA, we can generally see clearer groupings of the data than in the plots of the original data. For all of the datasets, the data reconstruction is working as expected, since the plots of the reconstructed data look identical to those of the original data.

In the Pen-based dataset (Figure 4), the transformed data is in a ball-like shape where the classes can be more clearly identified than in the original plot and all the points that were stacked on the edges of the plot are now grouped with the others in the center.



[H]

Figure 4: Transformed data & Reconstructed Data Pen-based dataset using sklearn PCA

Upon transformation of the CMC data, the points were moved from the parallel plane configuration found in the original data to a cloud of points, however, the grouping of the data did not become clearer than in the original data. Even when rotating the plot during its creation, it was not possible to find clear groupings of data.

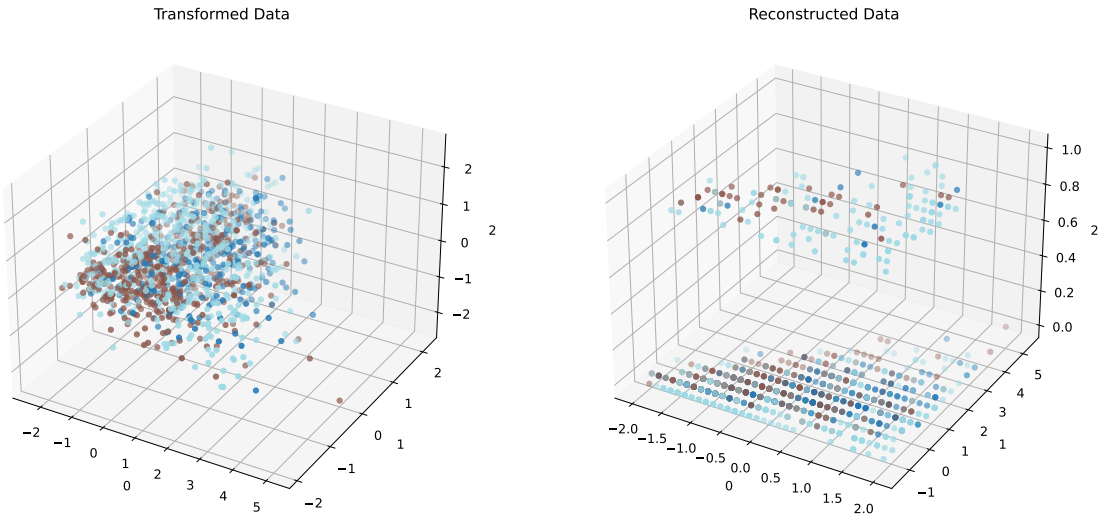


Figure 5: Transformed data & Reconstructed Data CMC dataset using sklearn PCA

During transformation of the Vowel dataset, the points became more spaced out than they were in the plot of the original data. Clusters of points are still identifiable, and the data retained its round shape, though is no longer elliptical. Despite the greater spacing, the positioning of the data points would still make it difficult to draw hyperplanes to definitively separate the data.

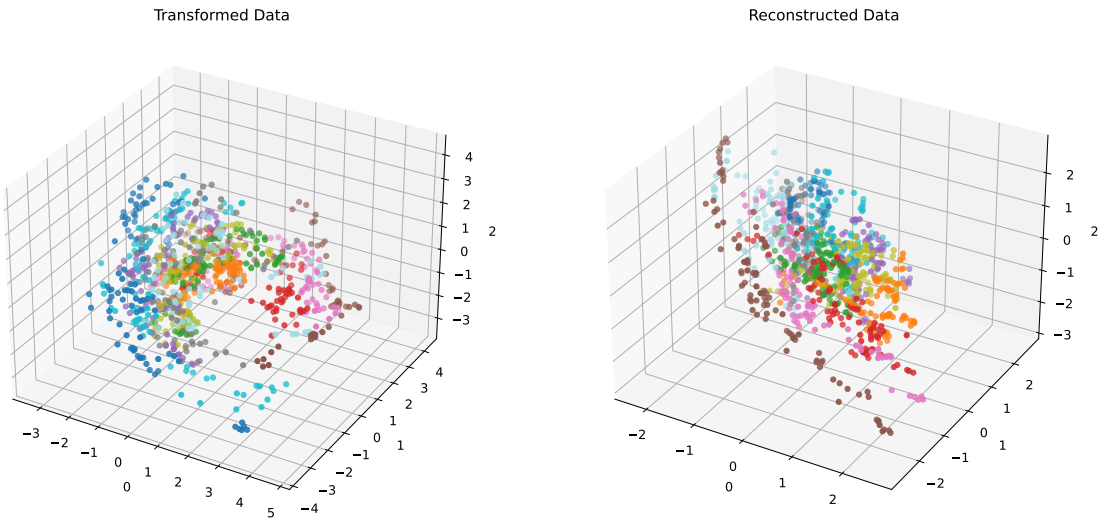


Figure 6: Transformed data & Reconstructed Data Vowel dataset using sklearn PCA

3.1.2 sklearn Incremental PCA

As in the previous sklearn PCA, the data reconstruction is working as expected, since the plots of the reconstructed data look identical to those of the original data. In general, the transformed data across all the datasets using Incremental PCA produces similar shapes as those generated in the original sklearn PCA, usually rotated as well.

The transformed data from the Pen-based dataset (Figure 7) represents the same ball-like shape seen in the original sklearn PCA. We can also still clearly see the groups.

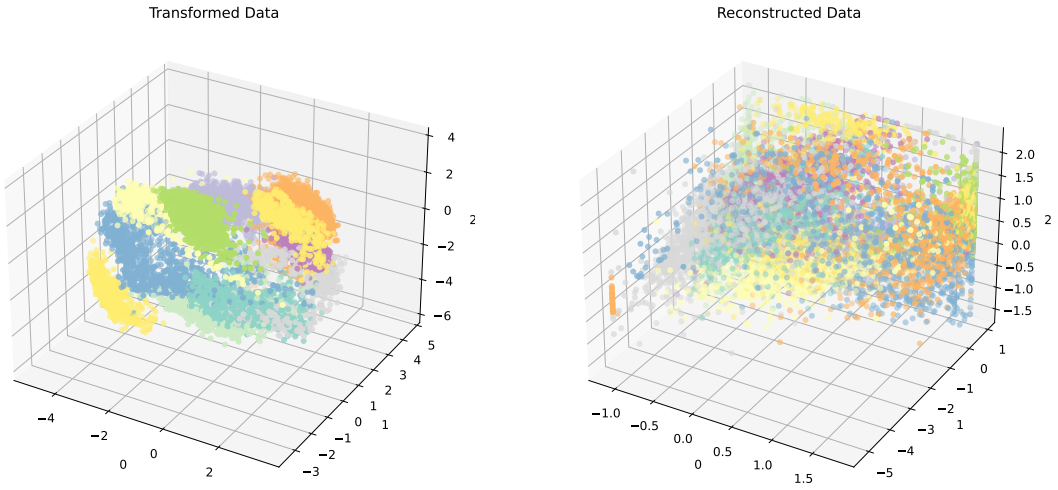


Figure 7: Transformed data & Reconstructed data Pen-based dataset using sklearn Incremental PCA

The transformed data for CMC dataset (Figure 8) is again a cloud-like shape and we cannot differentiate the classes.

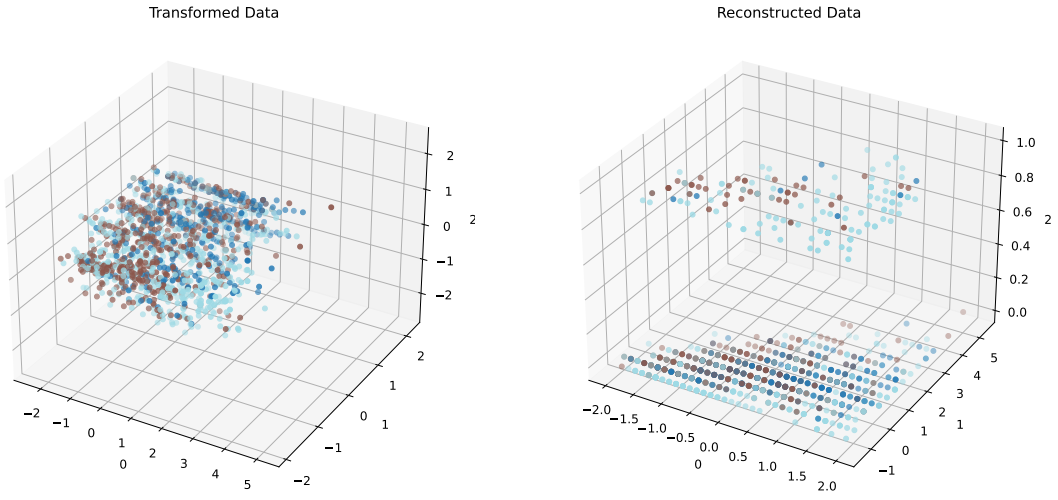


Figure 8: Transformed data & Reconstructed Data CMC dataset using sklearn Incremental PCA

The transformed data for the Vowel dataset (Figure 9) looks identical to the transformed data generated from the original sklearn PCA. This behavior is unexpected, given that with the Pen-based and CMC datsets, there were noticeable differences between the two plots.

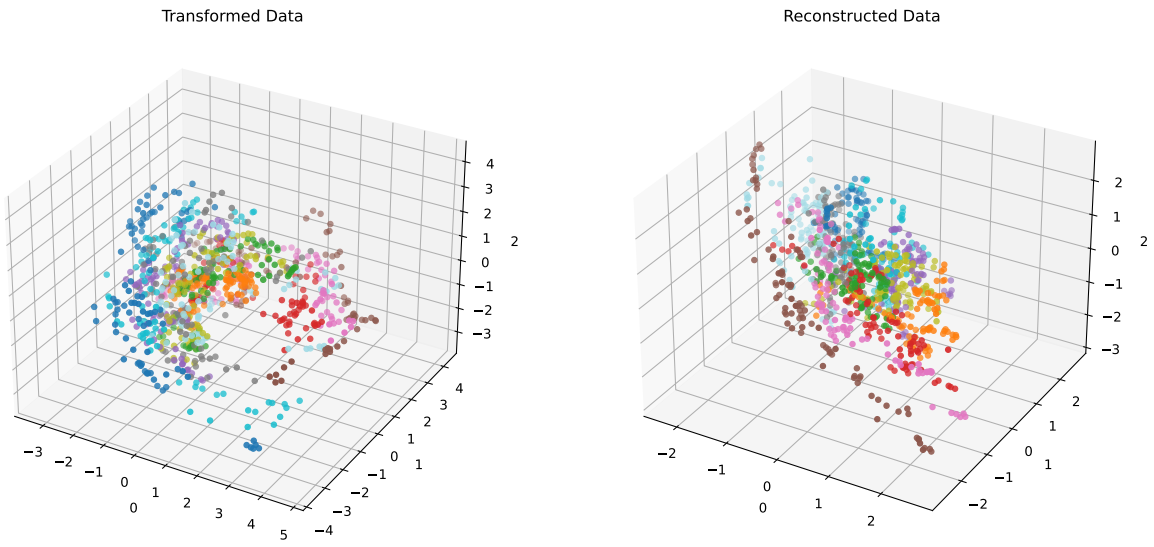


Figure 9: Transformed data & Reconstructed Data Vowel dataset using sklearn Incremental PCA

3.1.3 OPCA

The following plots represent the transformed data generated from OPCA. Again, the reconstruction is working as expected, since the plot of reconstructed data matches the plot of the original data, which indicates that our method of calculating the eigenvectors is correct. Additionally, the general shapes of the transformed data is expected based off the results from sklearn, as the plots are very similar.

The transformed data for the Pen-based dataset (Figure 10) presents the same shape as the sklearn PCA implementations, but as viewed from a different perspective, with clear groupings still maintained. The point of view we present here makes it easier to differentiate between the classes than can be done in the visualizations of the previous implementations.

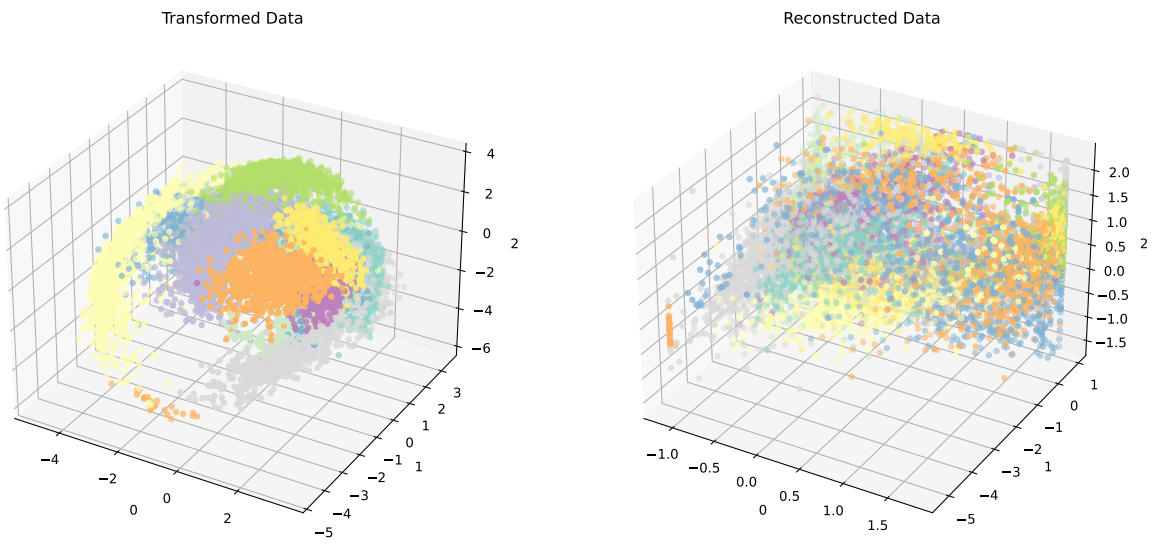


Figure 10: Transformed data & Reconstructed Data Pen-based dataset using OPCA

As with the two sklearn PCA implementations, the transformed data represents a cloud-like shape, with groups not clearly identifiable.

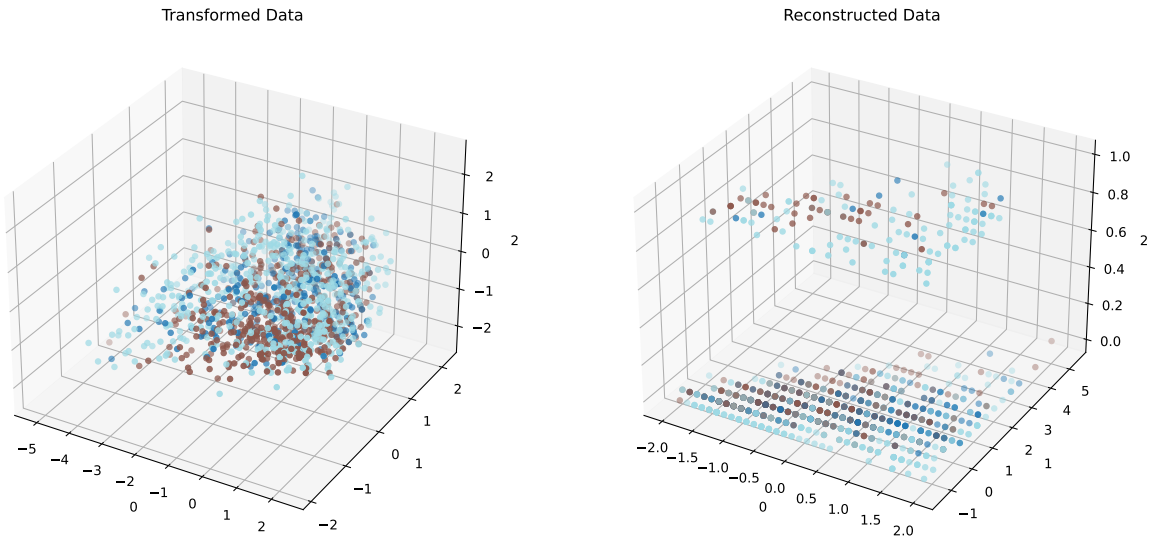


Figure 11: Transformed data & Reconstructed Data CMC dataset using OPCA

Again similar to the two sklearn PCAs, our transformed data for the Vowel dataset is in a more spread out round shape as compared with the original data.

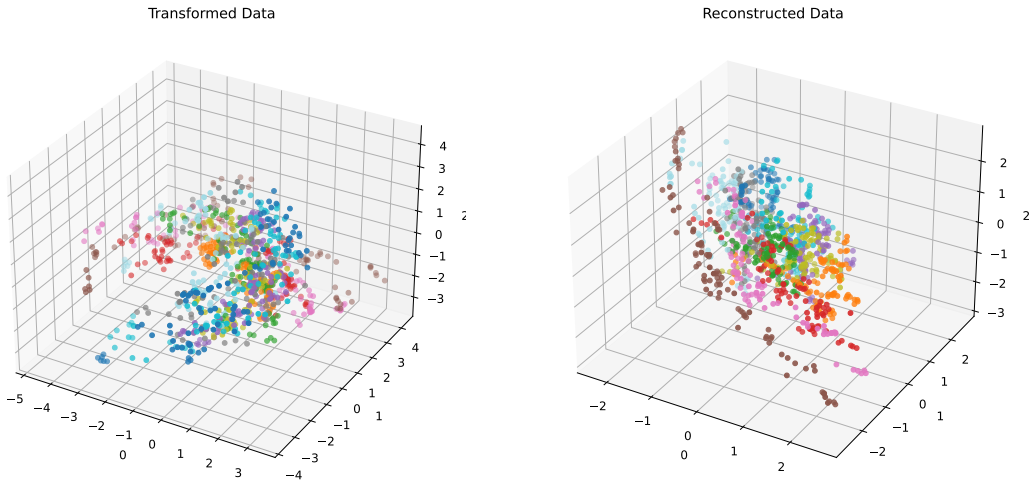


Figure 12: Transformed data & Reconstructed data Vowel dataset using OPCA

3.1.4 Feature Agglomeration

Using the sklearn imlementation of Feature Agglomeration, we do not acheive the best results for data visualization, where the objective is to be able to clearly define classes. We speculate that Feature Agglomeration works better with numerical datasets than with mixed and categorical datasets, since clearer clusters can be seen with our numerical Pen-based datasaet (Figure 13) than with our mixed CMC and Vowel datasets (Figure 14 and Figure 15). However, even with the numerical data, the performance is not optimal. For example, in the Pen-based plot, there are clearer groups than in the original data, however, they are not clear enough to definitively create clear groups. There is also still a lot of data on the outer boundaries of our plot, as seen in the original data.

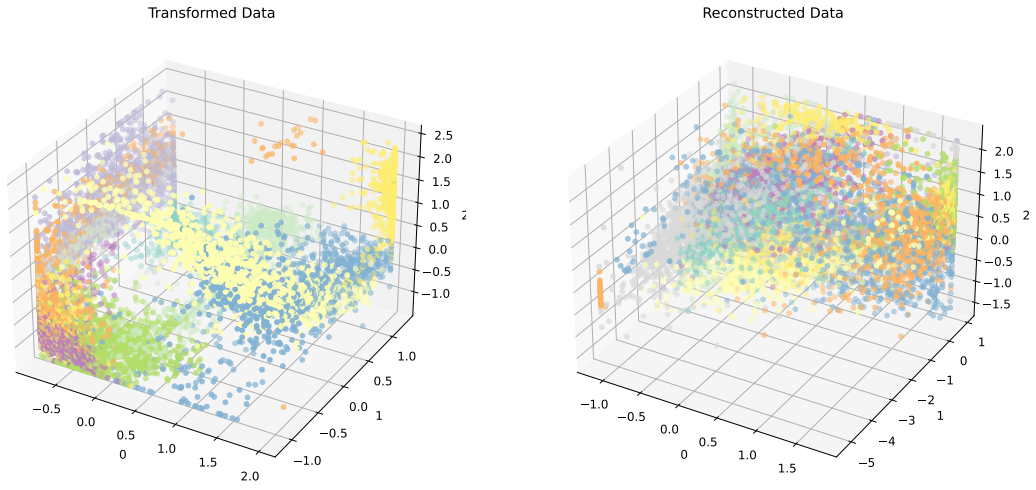


Figure 13: Transformed data & Reconstructed Data Pen-based dataset using sklearn Feature Agglomeration

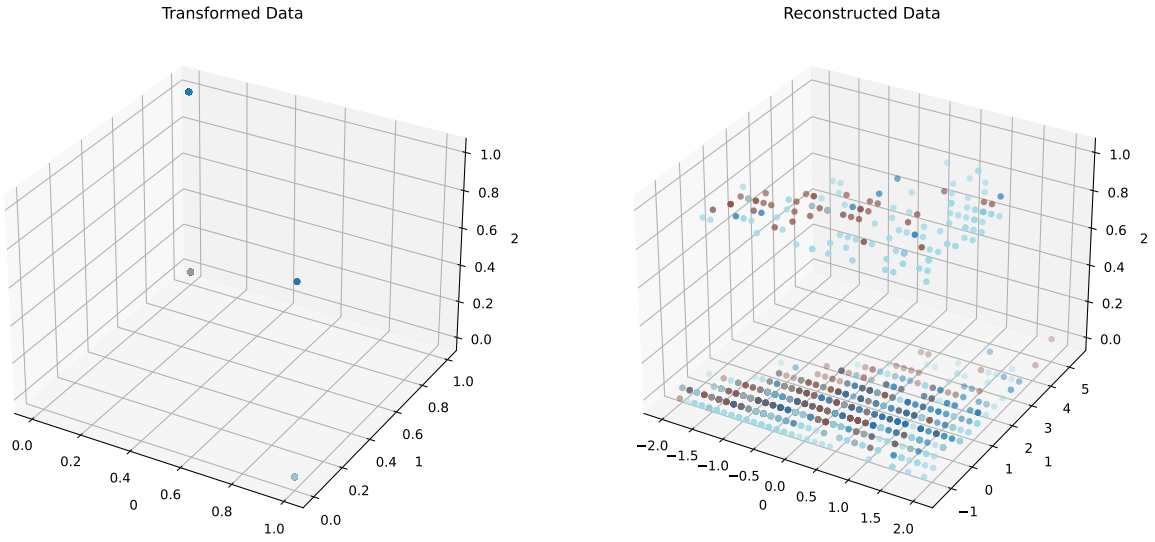


Figure 14: Transformed data & Reconstructed Data CMC dataset using sklearn Feature Agglomeration

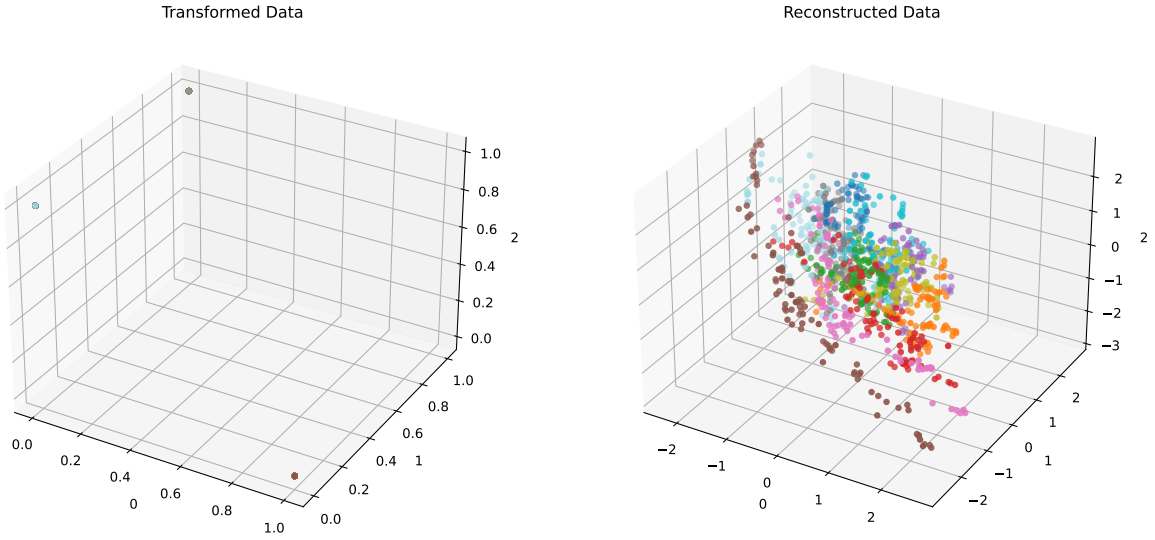
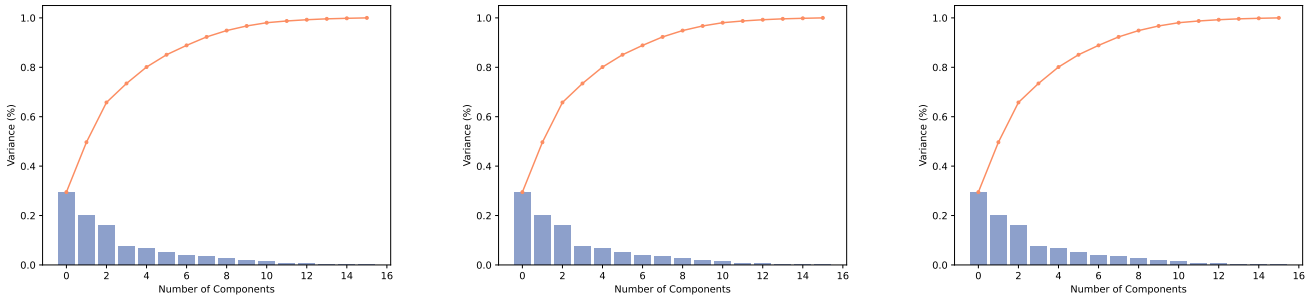


Figure 15: Transformed data & Reconstructed Data Vowel dataset using sklearn Feature Agglomeration

3.2 Explained Variance

Below are plots to demonstrate the total explained variance. These plots show that information is lost when the number of components is reduced to too low a number, and is therefore useful in helping us determine the number of components we should define when we run our clustering algorithms in Experiment 2. These plots also confirm that

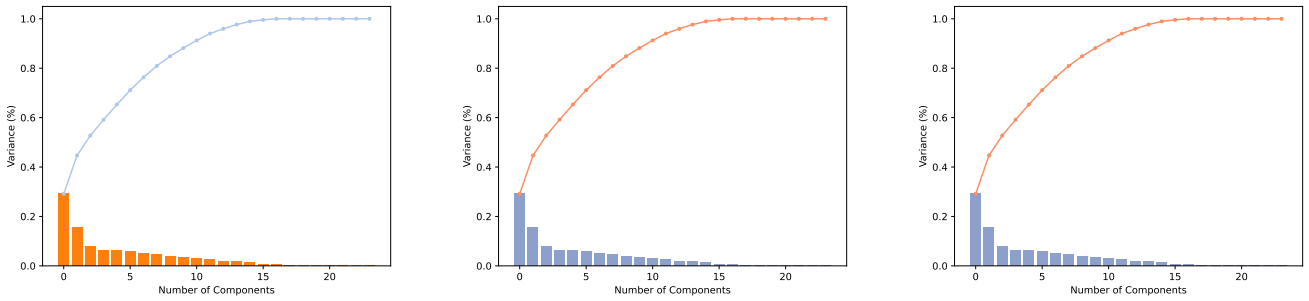
our OPCA is working in the expected way, since the plots are identical to those from the two sklearn implementations.



(a) OPCA (b) sklearn PCA (c) sklearn Incremental PCA

Figure 16: Explained variance plots for Pen-based dataset

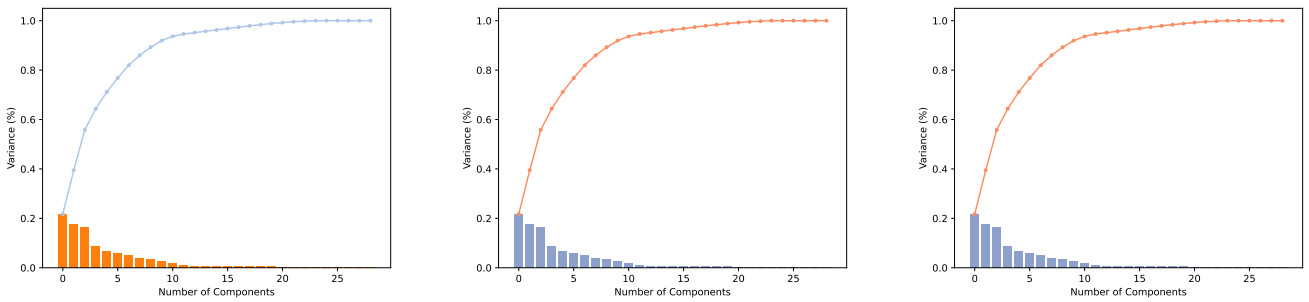
In **??** above, we can see that an ideal number of components could be 9, since there is a limited amount of information lost and we are still meaningfully reducing the number of components.



(a) OPCA (b) sklearn PCA (c) sklearn Incremental PCA

Figure 17: Explained variance plots for CMC dataset

In Figure 17 above, we can see that an ideal number of components could be 13, since again there is a limited amount of information lost.



(a) OPCA (b) sklearn PCA (c) sklearn Incremental PCA

Figure 18: Explained variance plots for Vowel dataset

Finally, in Figure 18 above, we can see that an ideal number of components could be 12.

In all the above cases, we tried to keep the total explained variance above 90%.

3.2.1 Experiment 1 Analysis

Based on the plots of transformed data from our experiments with PCA, we found that PCA is generally helpful for knowing the underlying information in the dataset. However, it depends on the nature of data. For example, the transformed data from the CMC dataset (Figure 11) does in fact regroup the data, however, it does not create clearly identifiable groups of data, so does not yield further information. On the other hand, the transformed data from the Pen-based dataset (Figure 10) does show much clearer groupings and therefore gives us more information about the dataset. The other dimensionality reduction technique we tested, Feature Agglomeration, provides less information about the dataset than PCA does. In situations where the data is purely numerical, it does provide some limited information about the clustering of data points, however, these clusters are not clear. An example of this can be seen with our Pen-based data transformation in Figure 13. When it comes to mixed and categorical datasets, there is a large amount of information loss, and there is very little data left from which to draw any meaningful conclusions about the data. This can be seen in our transformed data for the mixed Vowel dataset in Figure 12, where there are only three points in the Transformed Data plot. Despite this data loss, the algorithm is still able to reconstruct the data correctly. Regardless of whether or not our dimensionality reduction provided helpful information, it was still always possible to perform the dimensionality reduction with our datasets, both with PCA and with Feature Agglomeration. This is because we were able to adapt even our categorical data to be numerical data and allow for processing by the dimensionality reduction functions. Using the explained variance ratio values and the plots of total explained variance, we were able to find an ideal number of principal components for PCA or the ideal number of clusters for Feature Agglomeration. This reduced the Pen-based dataset from 16 to 9 features, the CMC dataset from 24 to 13 features, and the Vowel dataset from 29 to 12 features. However, as mentioned, Feature Agglomeration was not as informative as PCA, even when manipulating the linkage and affinity parameters in FA. The implementation of PCA is not overly complicated, especially when having libraries, such as NumPy and SciPy to do matrix calculations, which would otherwise be the most complex part. We were sure to explicitly calculate the eigenvalues, eigenvectors, covariance matrix, and explained variance ratio to be able to extract this information for comparison against the sklearn PCAs. The results obtained from OPCA are almost exactly the same as those obtained from the sklearn implementation of PCA. The only difference between the plots of transformed data is that the rotations are different; however, the shape is the same. These rotations are likely due to the way that SciPy is calculating the eigenvectors and eigenvalues.

3.3 Experiment 2 Results

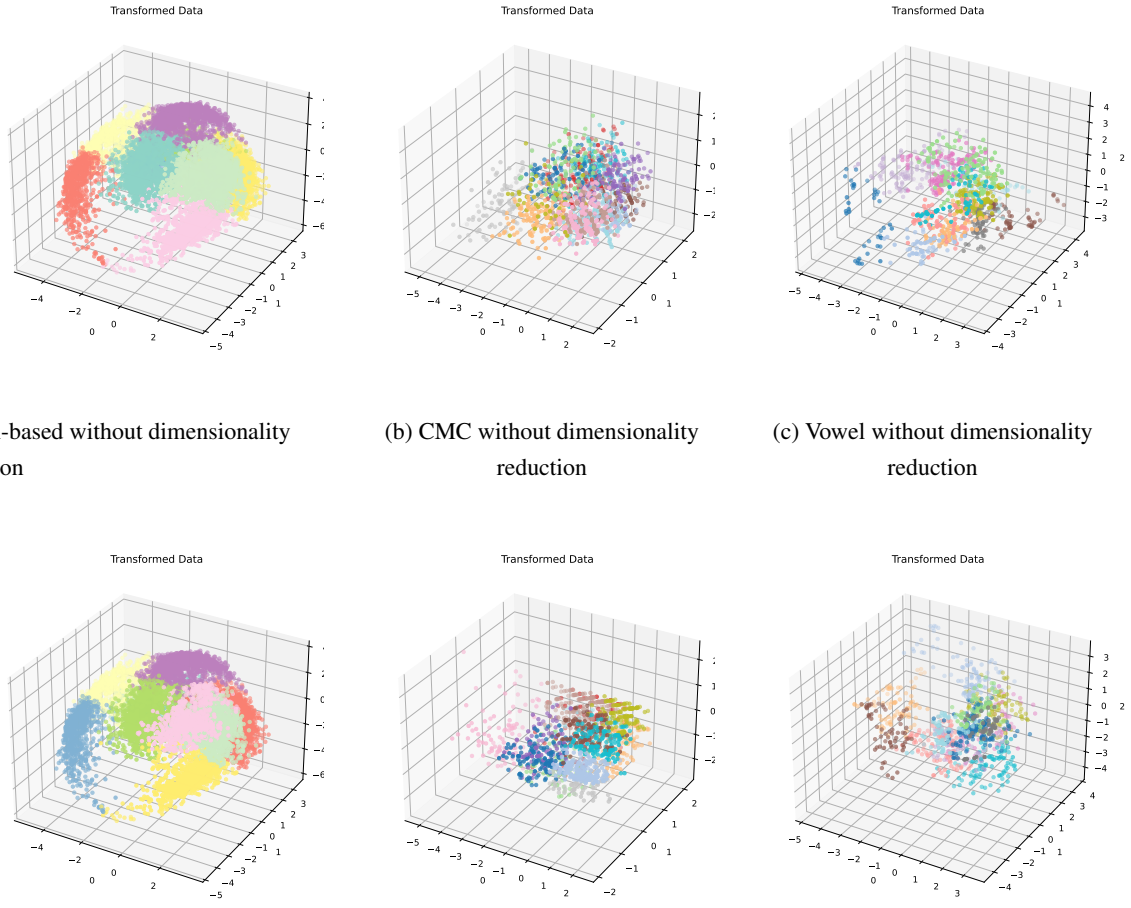
For Experiment 2, we first applied either OPCA, FA, or no dimensionality reduction technique to our dataset. We then clustered the resulting transformation (or the original data, in the even that no dimensionality reduction technique was applied) using either k-means or Agglomerative Clustering. Finally, we visualized the results using either OPCA or t-SNE.

3.3.1 PCA

In this section, we will detail the results of using PCA as the dimensionality reduction technique, and compare this directly with the results from using no dimensionality reduction technique, in combination with each of the clustering and visualization methods.

3.3.1.1 OPCA - K-means - OPCA

The plots below in Figure 19 show that with the Pen-based dataset in plots (a) and (d), the shape is almost exactly the same, meaning that in this numerical dataset, OPCA dimensionality reduction had little effect. The other datasets have different shapes before and after dimensionality reduction, showing that in these mixed datasets, OPCA had a greater effect.



(a) Pen-based without dimensionality reduction

(b) CMC without dimensionality reduction

(c) Vowel without dimensionality reduction

(d) Pen-based with PCA dimensionality reduction

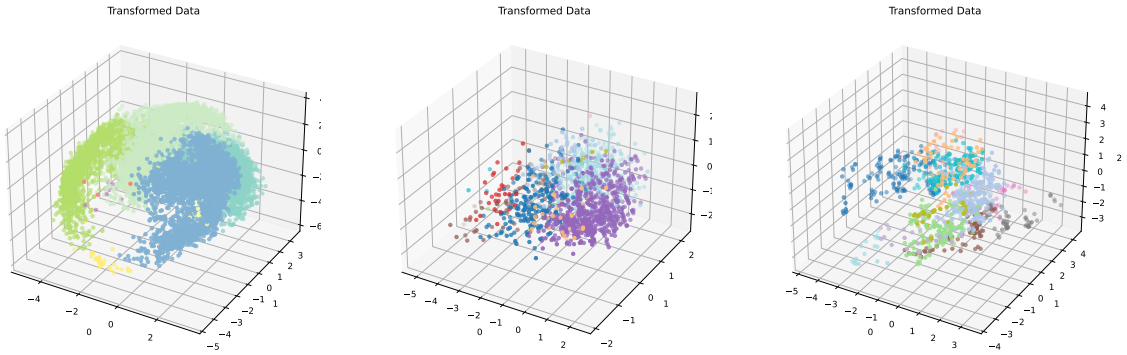
(e) CMC with PCA dimensionality reduction

(f) Vowel with PCA dimensionality reduction

Figure 19: Transformed data for Pen-based dataset

3.3.1.2 PCA - Agglomerative Clustering - PCA

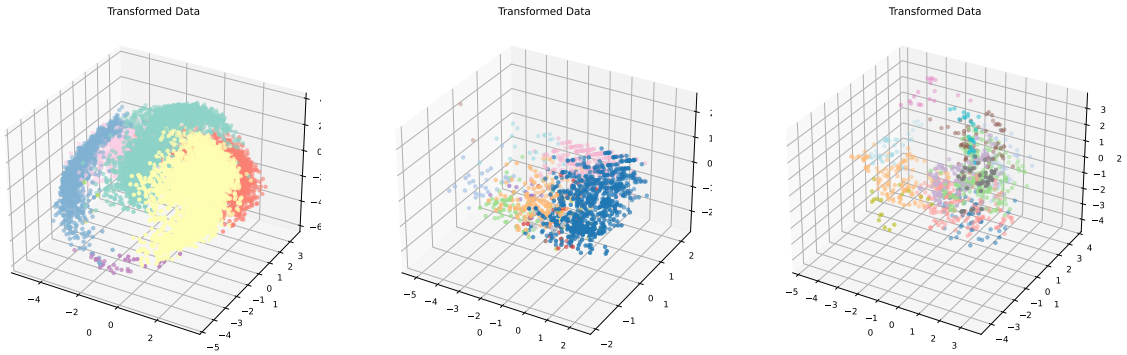
The plots below in Figure 20 show again that with the Pen-based dataset in plots (a) and (d), the shape is almost exactly the same, meaning that in this numerical dataset, OPCA dimensionality reduction had little effect. The other datasets have different shapes before and after dimensionality reduction, showing that in these mixed datasets, OPCA had a greater effect.



(a) Pen-based without dimensionality reduction

(b) CMC without dimensionality reduction

(c) Vowel without dimensionality reduction



(d) Pen-based with PCA dimensionality reduction

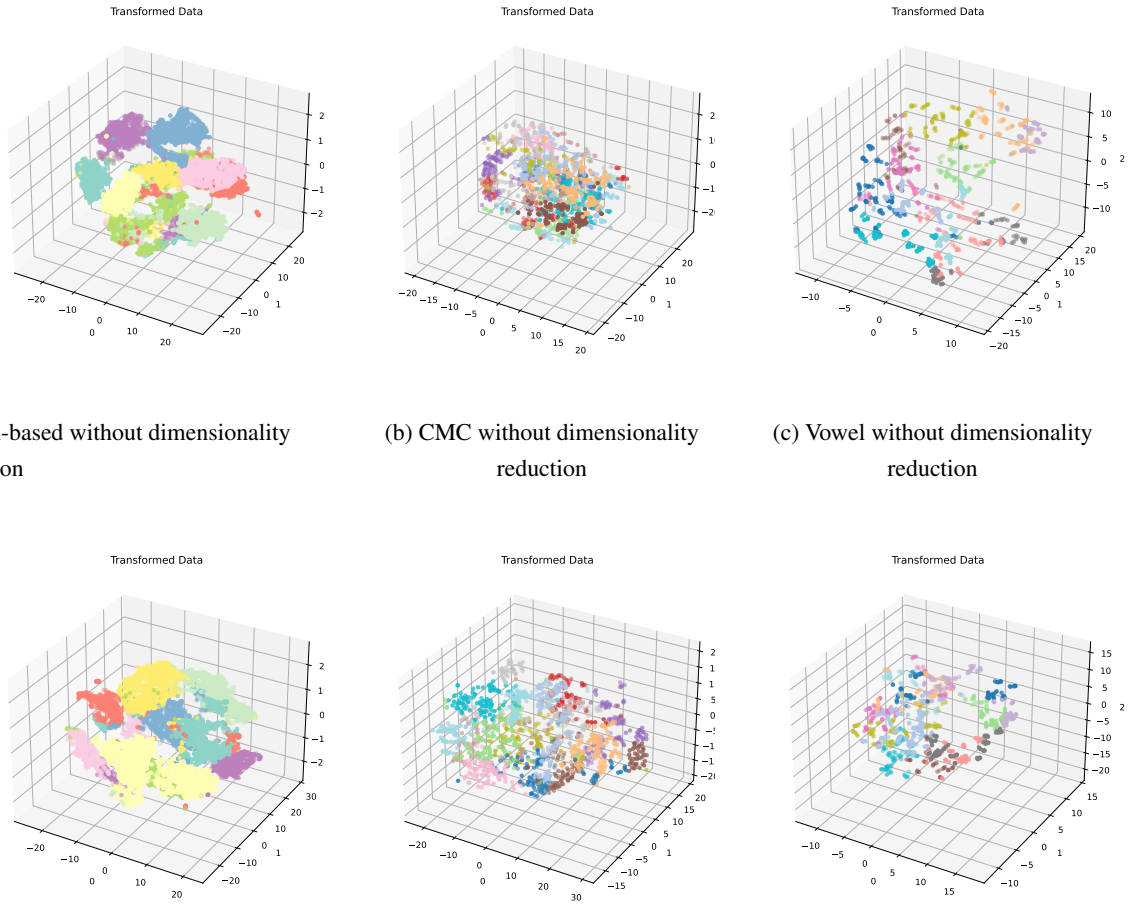
(e) CMC with PCA dimensionality reduction

(f) Vowel with PCA dimensionality reduction

Figure 20: Transformed data for PCA - Agglomerative Clustering - PCA

3.3.1.3 PCA - K-means - t-SNE

The plots below in Figure 21 show that in each dataset, the plots have different shapes with and without dimensionality reduction, showing that OPCA had an effect on the overall data transformation.



(a) Pen-based without dimensionality reduction

(b) CMC without dimensionality reduction

(c) Vowel without dimensionality reduction

(d) Pen-based with PCA dimensionality reduction

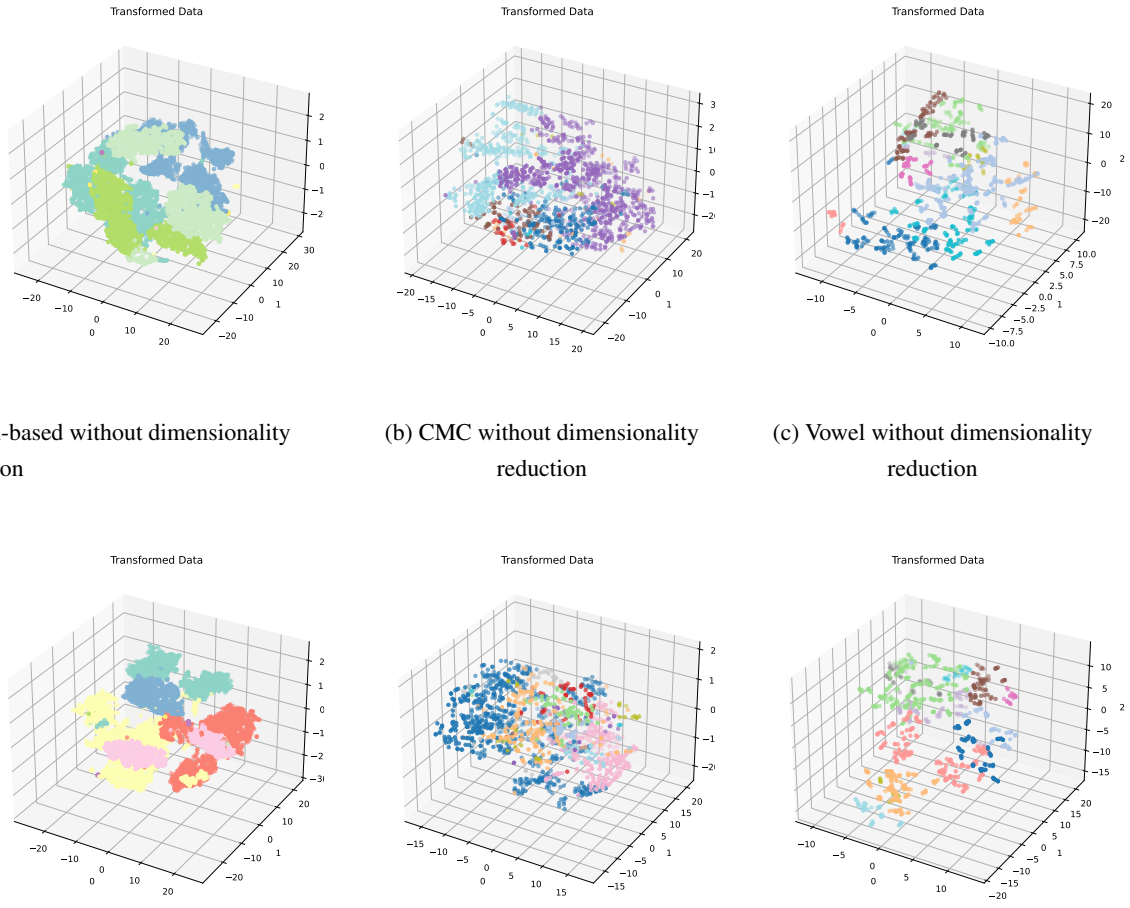
(e) CMC with PCA dimensionality reduction

(f) Vowel with PCA dimensionality reduction

Figure 21: Transformed data for PCA - K-means - t-SNE

3.3.1.4 PCA - Agglomerative Clustering - t-SNE

The plots below in Figure 22 show that in each dataset, the plots have slightly different shapes with and without dimensionality reduction, showing that OPCA had an effect on the overall data transformation.



(a) Pen-based without dimensionality reduction

(b) CMC without dimensionality reduction

(c) Vowel without dimensionality reduction

(d) Pen-based with PCA dimensionality reduction

(e) CMC with PCA dimensionality reduction

(f) Vowel with PCA dimensionality reduction

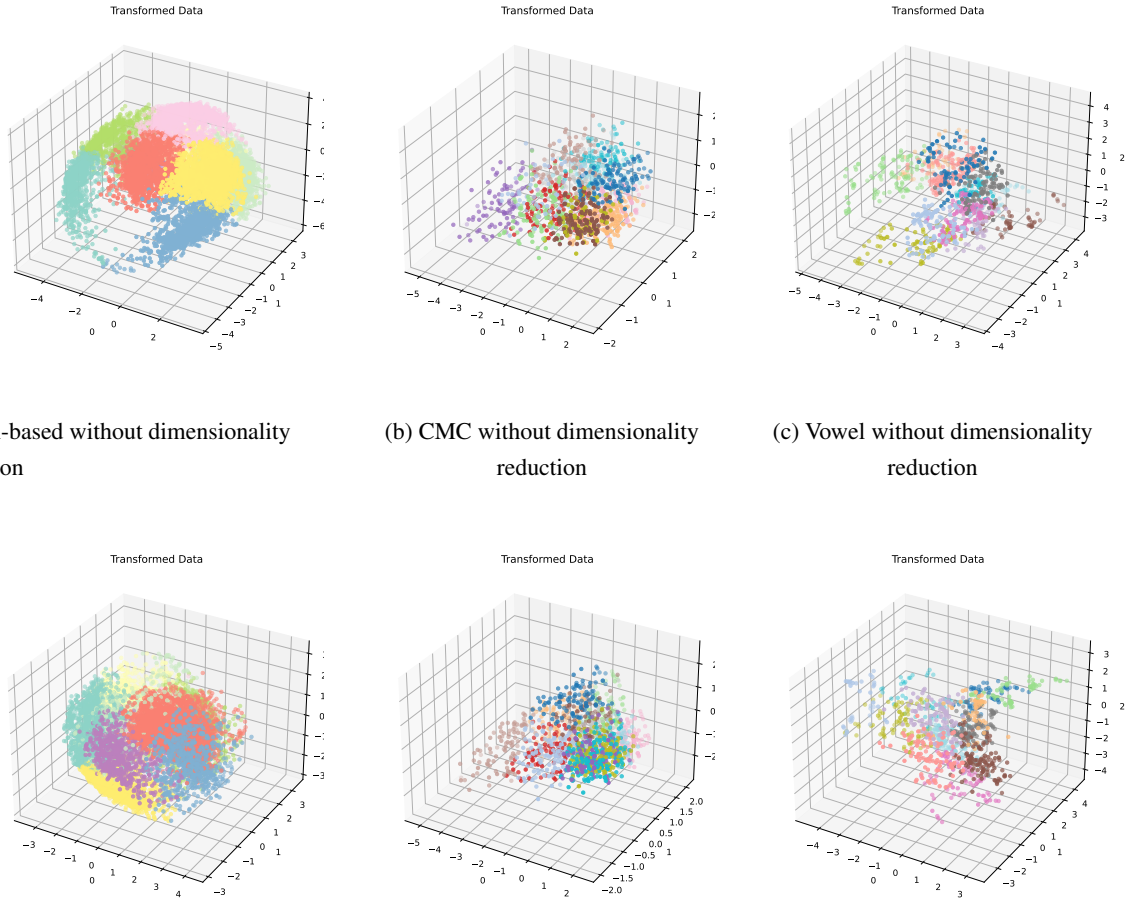
Figure 22: Transformed data for PCA - Agglomerative Clustering - t-SNE

3.3.2 Feature Agglomeration

In this section, we will detail the results of using FA as the dimensionality reduction technique, and compare this directly with the results from using no dimensionality reduction technique, in combination with each of the clustering and visualization methods.

3.3.2.1 FA - K-means - PCA

The plots below in Figure 23 show that in each dataset, the plots have different shapes with and without dimensionality reduction, showing that OPCA had an effect on the overall data transformation.



(a) Pen-based without dimensionality reduction

(b) CMC without dimensionality reduction

(c) Vowel without dimensionality reduction

(d) Pen-based with FA dimensionality reduction

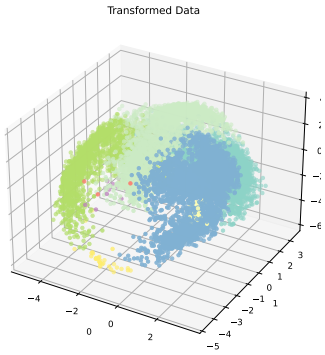
(e) CMC with FA dimensionality reduction

(f) Vowel with FA dimensionality reduction

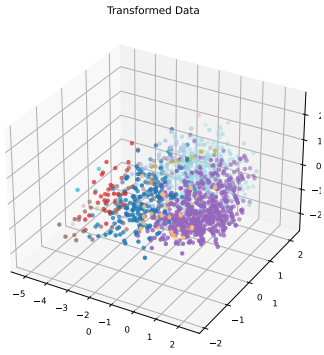
Figure 23: Transformed data for FA - K-means - PCA

3.3.2.2 FA - Agglomerative Clustering - PCA

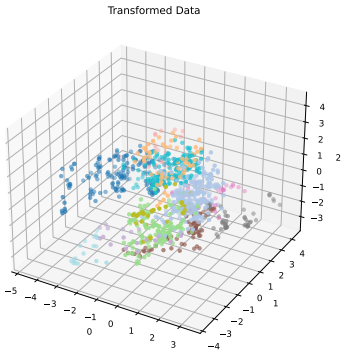
The plots below in Figure 24 show that in each dataset, the plots have slightly different shapes with and without dimensionality reduction, showing that OPCA had an effect on the overall data transformation.



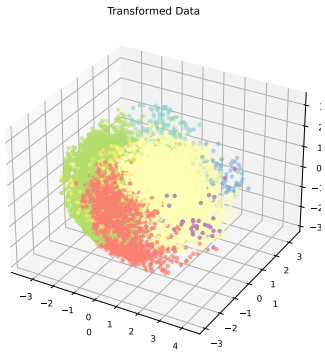
(a) Pen-based without dimensionality reduction



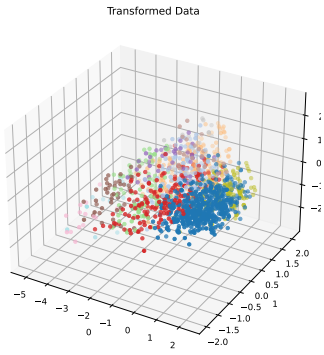
(b) CMC without dimensionality reduction



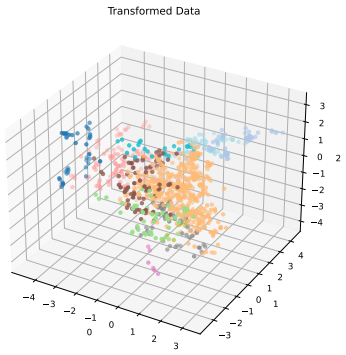
(c) Vowel without dimensionality reduction



(d) Pen-based with FA dimensionality reduction



(e) CMC with FA dimensionality reduction

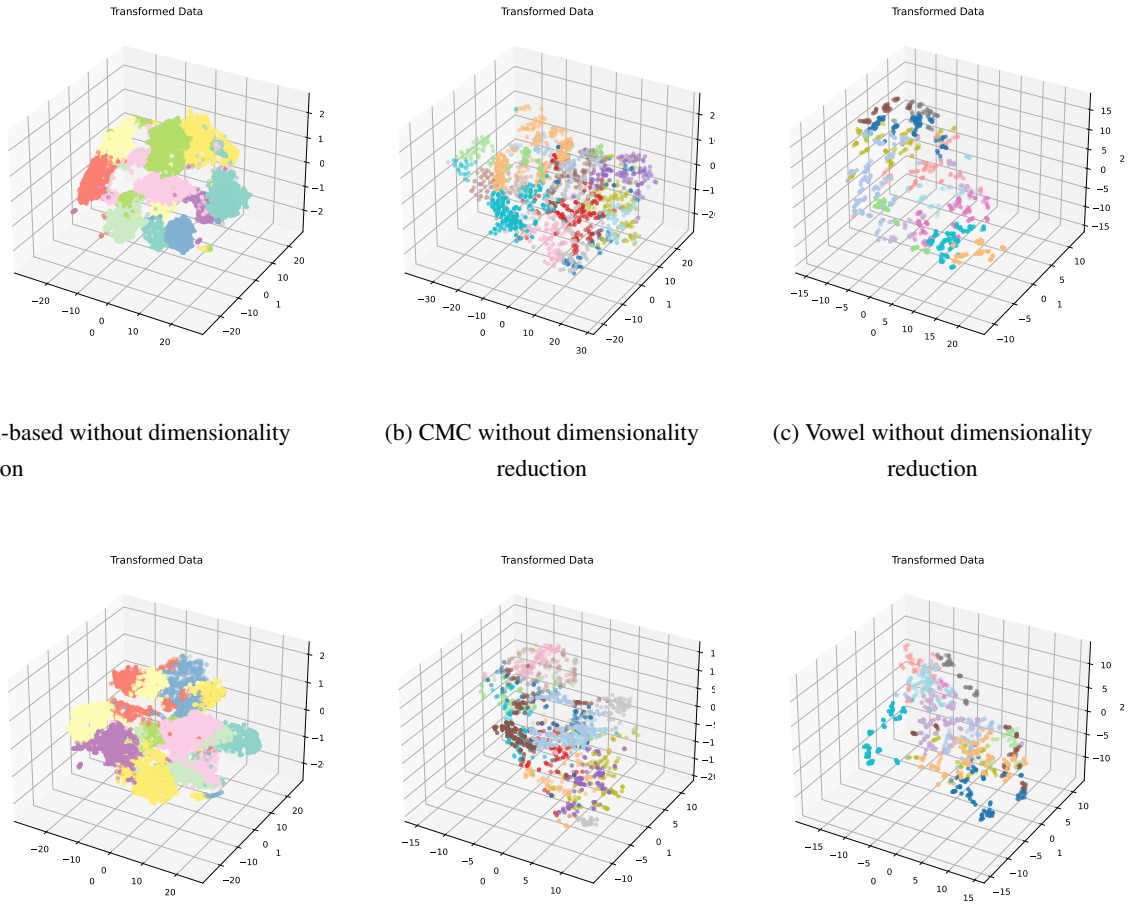


(f) Vowel with FA dimensionality reduction

Figure 24: Transformed data for FA - Agglomerative Clustering - PCA

3.3.2.3 FA - K-means - t-SNE

The plots below in Figure 25 show that in each dataset, the plots have different shapes with and without dimensionality reduction, showing that OPCA had an effect on the overall data transformation.



(a) Pen-based without dimensionality reduction

(b) CMC without dimensionality reduction

(c) Vowel without dimensionality reduction

(d) Pen-based with FA dimensionality reduction

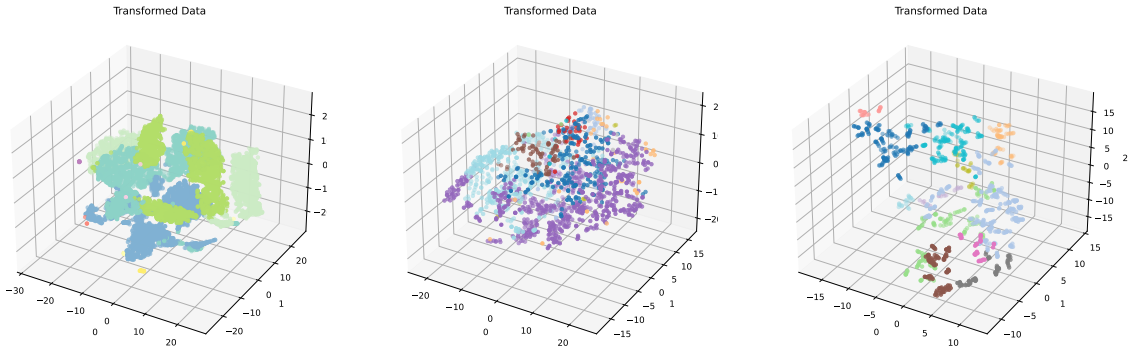
(e) CMC with FA dimensionality reduction

(f) Vowel with FA dimensionality reduction

Figure 25: Transformed data for FA - K-means - t-SNE

3.3.2.4 FA - Agglomerative Clustering - t-SNE

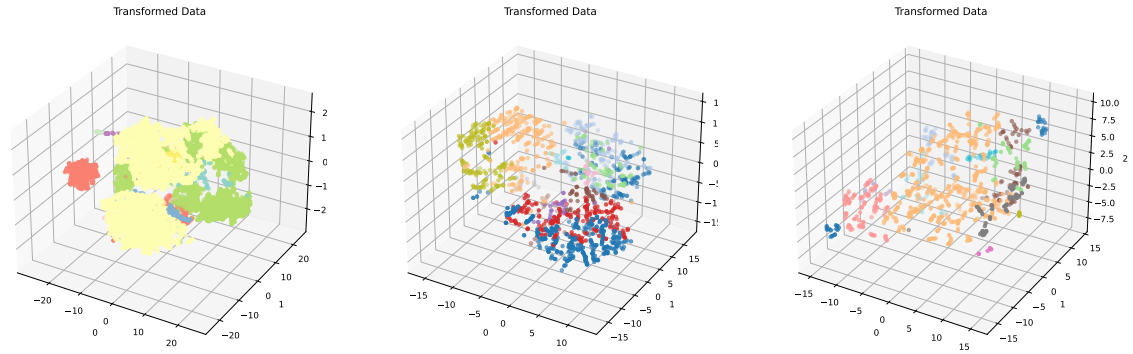
The plots below in Figure 26 show that in each dataset, the plots have different shapes with and without dimensionality reduction, showing that OPCA had an effect on the overall data transformation.



(a) Pen-based without dimensionality reduction

(b) CMC without dimensionality reduction

(c) Vowel without dimensionality reduction



(d) Pen-based with FA dimensionality reduction

(e) CMC with FA dimensionality reduction

(f) Vowel with FA dimensionality reduction

Figure 26: Transformed data for FA - Agglomerative Clustering - t-SNE

3.3.3 Experiment 2 Analysis

In most cases, applying a dimensionality reduction technique resulted in different clusters than those obtained from the nonreduced data. However, in our numerical Pen-based dataset, these differences tended to be much smaller, yielding very minor differences in the shape of the clusters. Larger differences tended to occur between the plots of our mixed CMC and Vowel datasets, but there were still some cases where the clustering was fairly similar. No significant reduction in time was observed. In general, PCA created more informative visualizations than t-SNE, with generally clearer groupings. Additionally, t-SNE visualizations typically took longer to generate than PCA visualizations.

4 Conclusion

From Experiment 1, we learned the process to successfully implement our own version of PCA, which was simpler than originally expected. We also saw that PCA tends to be better than Feature Agglomeration at providing meaningful information about the dataset, especially with mixed datasets, though neither is able to reliably provide a perfectly clear data transformation. From Experiment 2, we learned how to select the proper number of components to reduce based on the explained variance ratio. We also saw that it is not necessary to use a dimensionality reduction technique to achieve better clustering results. Finally, we learned more about how PCA and t-SNE can be used as visualization techniques.