Assignment 3

(svyas44@gatech.edu)

I. Datasets for Clustering and Dimensionality Reduction

Datasets Description:

a. Coil-20 Dataset: The Dataset contains 1440 image of 20 different categories of images Each image is of size 128 x 128. For simplicity, only coloured images were converted to grayscale.
b. WholeSale Customer Dataset: The dataset refers to clients of a wholesale distributor, consisting of 440 samples – relatively smaller than the previous Dataset. It includes the annual spending in monetary units on diverse product categories.

II. Experiments with Clustering

Clustering Methods used: Two Techniques – KMeans and Gaussian Mixture were used to conduct experiments on Clustering the above datasets. The simple reason of choosing these algorithms were easy interpretation of the results. Typically for image datasets, these clustering algorithms efficiently capture feature representation (colour histograms, texture features etc). The same have been widely preferred for transactional datasets as well to do cluster level analysis.

For the Dataset 1, since we had ground truth for the category of the image from the filename, we could do lot of interesting analysis. The Hypothesis was that the resultant cluster would result in 20 different categories, but the original vs resultant cluster distribution looks as follows in Cluster 1:

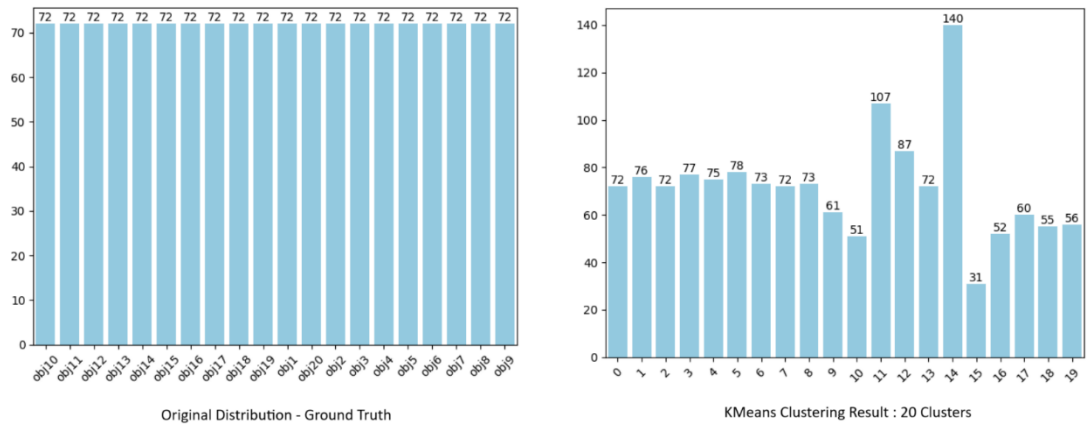

Figure 1: Ground Truth vs Resultant cluster comparison

We see good number of clusters where the count is same as ground truth. Upon analysis, it seems the images in these clusters also seem to be of the same category

which means that the algorithms are capturing correct features. For some clusters where count is either not exactly 72, we see that it might it clubs images together in same cluster based on similarity in orientation or shape as shown in Figure 2. This behaviour is observed in both KMeans and GMM clustering algorithm.

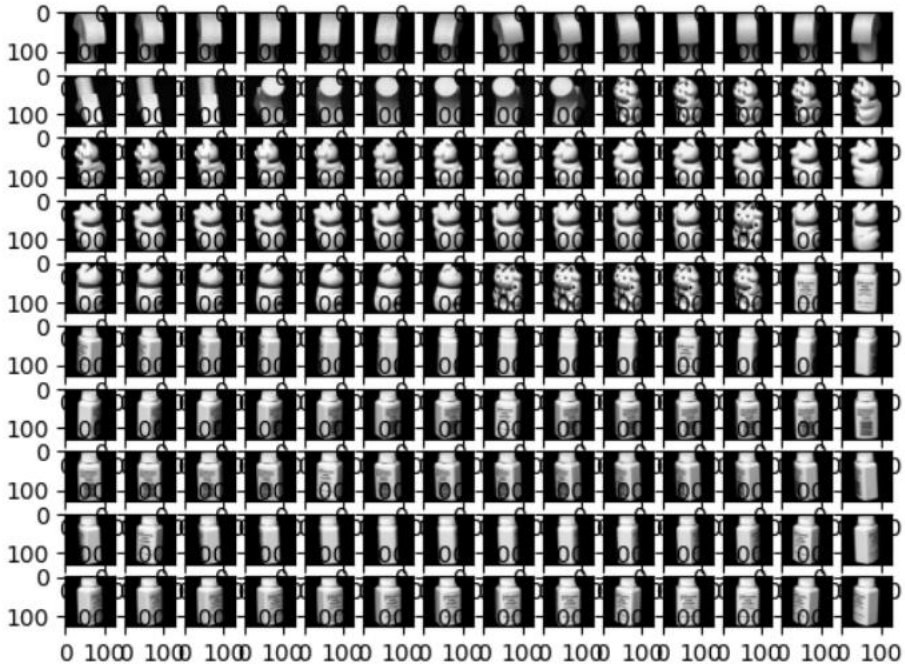

Figure 2: Cluster no 14 with multiple types of objects clubbed together due to some similarity in shape.

For the Dataset 2, the curves show clear convergence for optimal values of k as seen in Figure 4. This type of behaviour is not present in Dataset 1 since the Dataset is more complex and more cluster can isolate / model more information.

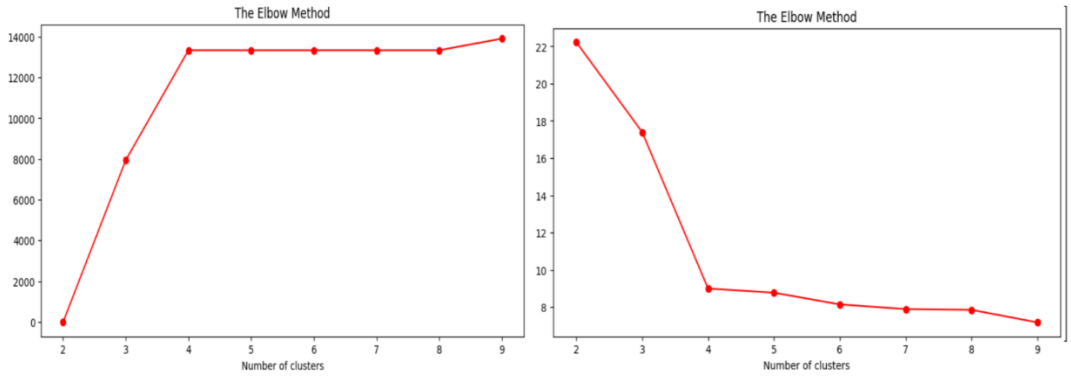

Figure 3: Log-likelihood and Inertia plot for GMM and KMeans plot show clear elbow at k=4

When we look for representative centroids produced during clustering, the centroids provide very good information on feature level, related to the shape of each object. This can be seen clearly in Figure 6. The objects might look like they are spinning but

the centroids try to capture different orientations of the object in the centroid, due to which there is a white pixel at each orientation.

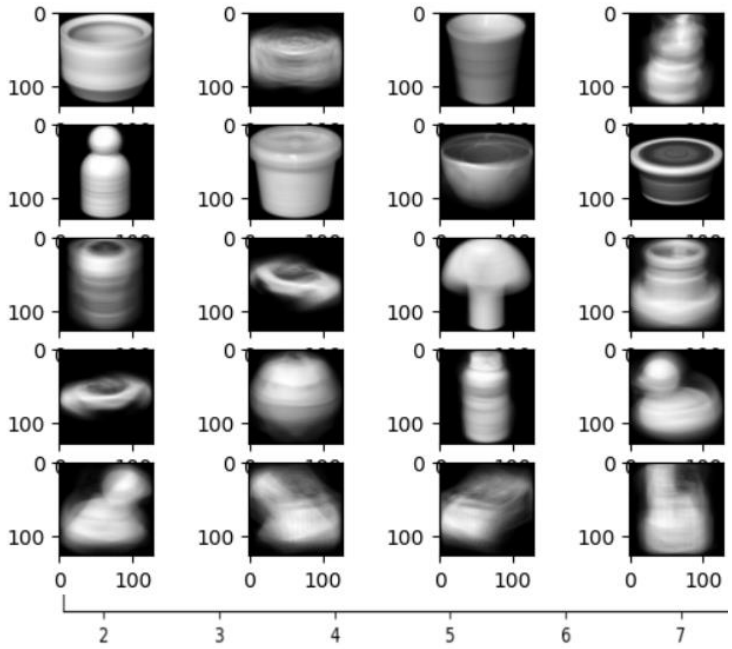

Figure 4: Images of centroids – similar images obtained GMM and KMeans

While the clusters obtained have very similar distribution for Dataset 1, we see some differences for Dataset 2 as seen in Figure 7. When we tried to analyse the reason by reducing the dimension using PCA, we see a very interesting pattern with the clusters as seen in Figure 8. This also validates the assumption of Gaussian Distributions, allowing for more flexible shapes rather than spherical assumed by KMeans. GMM also handles overlapping clusters by allowing datapoints to have membership in multiple Gaussian Distributions.

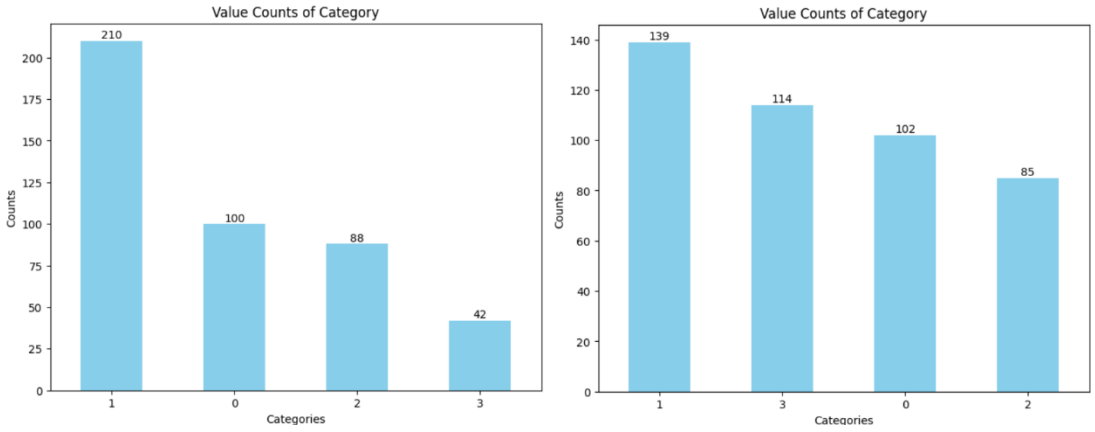

Figure 5: Cluster Distribution for GMM (left) vs KMeans (right)

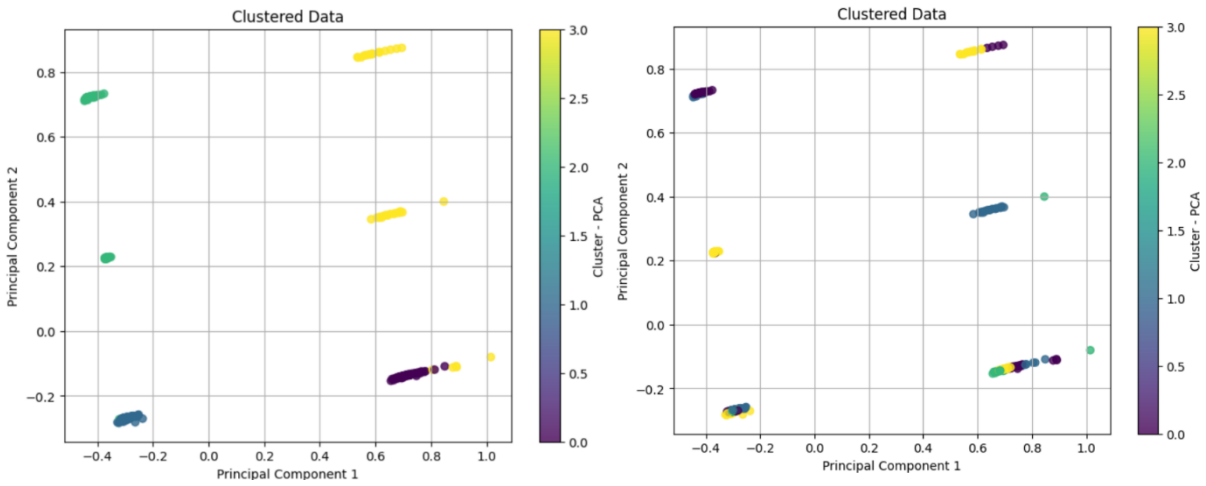

Figure 6: GMM (left) vs KMeans (right) – dimensions reduced to 2 using PCA.

III. Experiments with Dimensionality Reduction

As we reduce dimensionality, clear information can be seen for all the techniques. PCA captures dominant features from Dataset 1 – textures, edges, shapes, etc as seen in Figure 9.

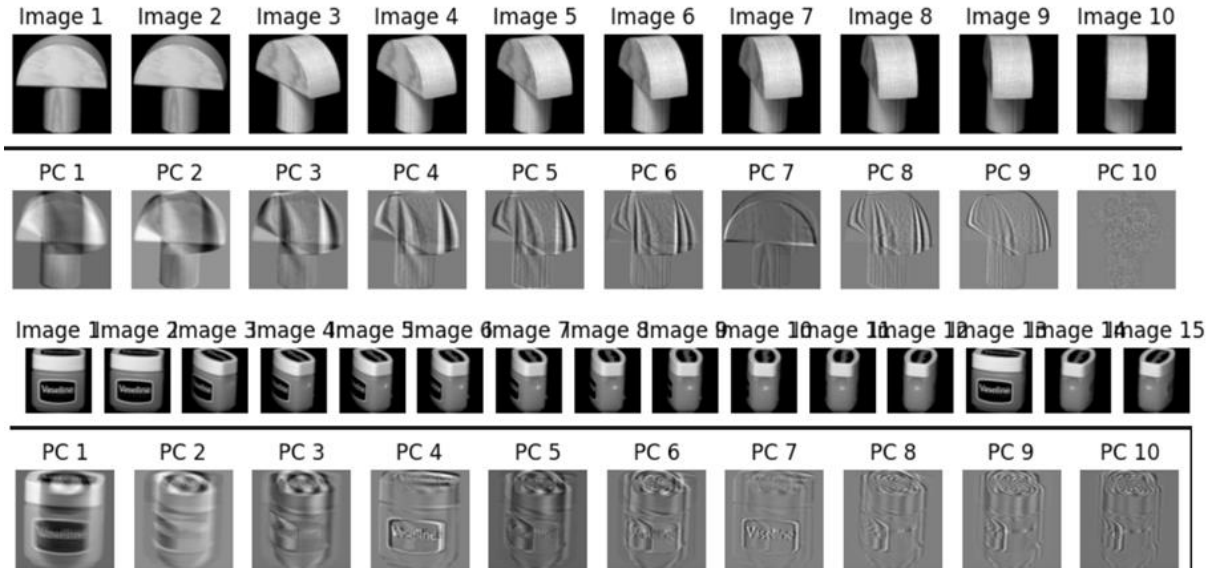

Figure 7: Sample Images and respective Principal Components

When flattened, the images have a feature size of 16384. PCA captures 98% of variance in less than 2% of the dimensions

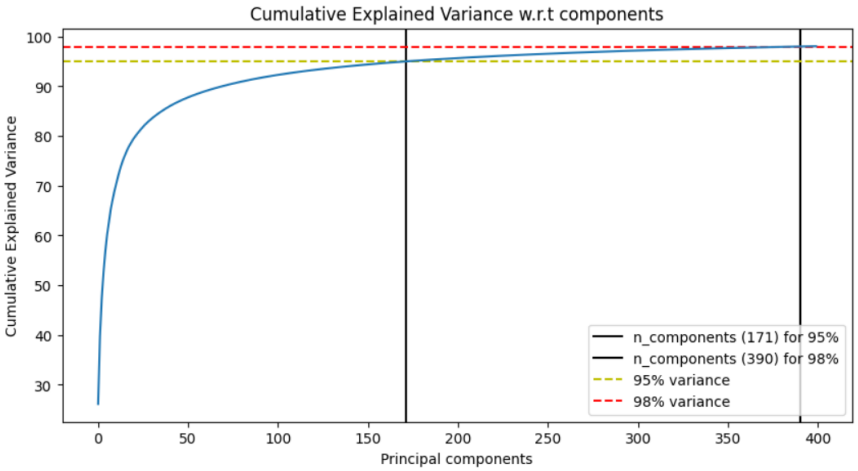

Figure 8: Variance Captured by principal components.

ICA captures the information in fewer components (<15) when measured for high kurtosis value. Both ICA and PCA seem to be good at capturing information from the images as it can be seen in reconstructed images – predominantly visible edges, texture, and clear separation of textures.

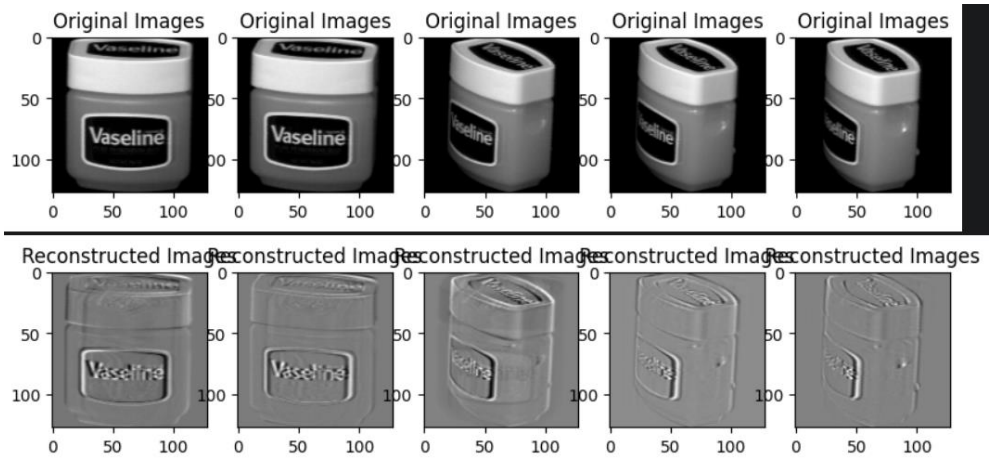

Figure 9: Reconstruction by ICA

When using Gaussian Randomized Projections, the reconstructed images show a very high information loss. Gaussian Randomized Projections seem to make sense only for very high dimensions. The MSE between original images and reconstructed images seem to very high for this method.

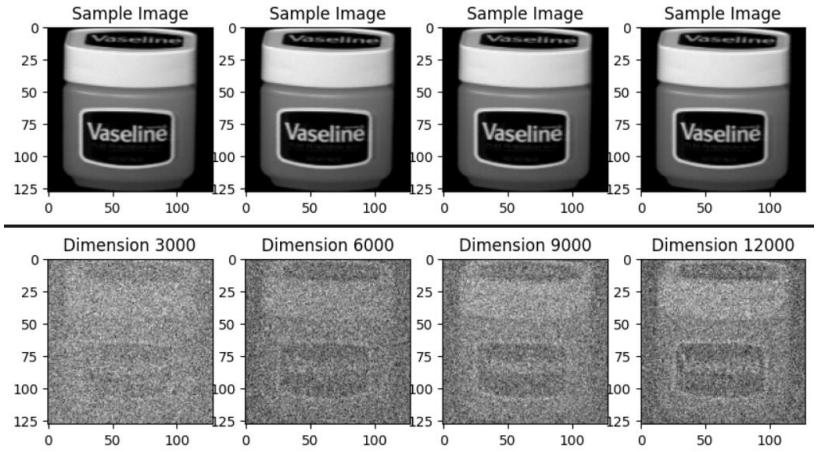

Figure 10: Reconstruction by Gaussian Randomized Projections

When using non-linear method, I used multi-dimensional scaling. While measuring stress, the convergence seems to be achieved on very less number of components:

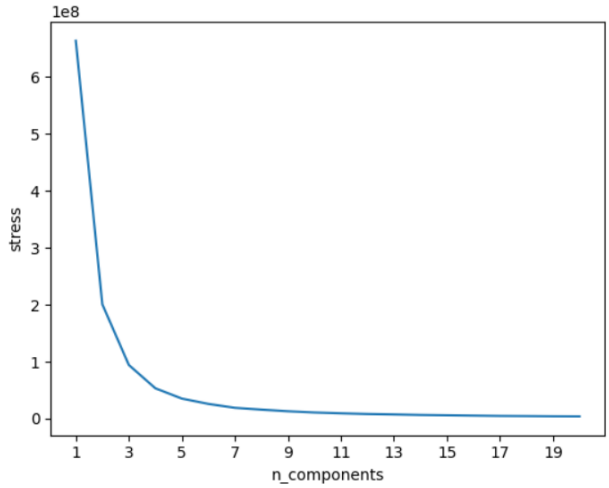

Figure 13: Stress values for MDS

IV. Clusters after Dimensionality Reduction

To measure how effective Dimensionality Reduction has been, we try to compare original clusters without dimensionality reduction vs with dimensionality reduction. PCA and GRP seem to have similar distribution, however ICA seems to worsen.

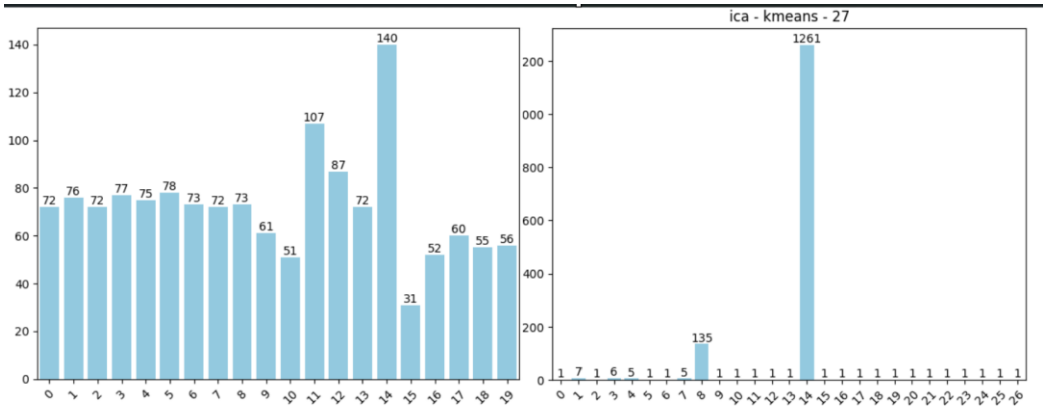

Figure 14: Cluster Distribution Without Dimensionality Reduction vs with Dimensionality Reduction with ICA

We also measure adjusted_rand_score, normalized_mutual_info_score, fowlkes_mallows_score as a supervised way to evaluate how close the clusters are to the original after dimension reduction. Summary of Results:

	Comparison with Ground Truth				**Comparison with Original Clusters**		
	ari	**nmi**	**fms**		**ari**	**nmi**	**fms**
PCA	0.601	0.784	0.624		0.79	0.894	0.802
ICA	0.015	0.0638	0.217		0.019	0.064	0.227
GRP	0.65	0.8	0.67		0.778	0.877	0.79
MDS	0.61	0.79	0.64		0.712	0.873	0.73

Table 1: Measuring performance of Dimensionality Reduction methods with KMeans Clustering: Dataset 1

	Comparison with Ground Truth				**Comparison with Original Clusters**		
	ari	**nmi**	**fms**		**ari**	**nmi**	**fms**
PCA	0.117	0.574	0.267		0.15	0.562	0.306
ICA	0.116	0.375	0.256		0.084	0.308	0.221
GRP	0.569	0.7833	0.5965		0.588	0.788	0.613
MDS	0.594	0.7979	0.62		0.617	0.822	0.64

Table 2: Measuring performance of Dimensionality Reduction methods with GMM Clustering: Dataset 1

	Comparison with Original Clusters - GMM				**Comparison with Original Clusters - Kmeans**		
	ari	**nmi**	**fms**		**ari**	**nmi**	**fms**
PCA	0.97	0.944	0.98		0.144	0.174	0.395
ICA	0.963	0.937	0.976		0.144	0.173	0.395
GRP	0.734	0.72	0.821		0.125	0.155	0.386
MDS	0.7725	0.774	0.851		0.144	0.173	0.395

Table 3: Measuring performance of Dimensionality Reduction methods with GMM and KMeans Clustering: Dataset 2

V. Neural Network Trained with and without Dimensionality Reduction (and Clustering information)

Dataset 1 was used to experiment with Neural Networks. The cluster information calculated in the above experiments was used as well, to see if they contribute any information. The performance figures of training a Neural Network looks as follows:

	Without Reducing Dimensionality		
	Recall	**Precision**	**f1**
Without using cluster information	0.9953	0.9953	0.9953
	Factor Analysis (Linear)		
	0.9675	0.9675	0.9675
	TSNE		
	0.023	0.023	0.023
After using cluster information	**Without Reducing Dimensionality**		
	0.986	0.986	0.986
	Factor Analysis (Linear)		
	0.9675	0.9675	0.9675
	TSNE		
	0.0463	0.0463	0.0463

VI. Conclusion

1. From results of Clustering with Dimensionality reduction, linear methods perform very well (PCA, GRP) as compared to the non-linear methods like MDS and TSNE. This indicates that there is clear linear separation amongst the clusters. Especially, results for PCA seem very impressive when we look at Dataset 2 and then at Dataset 1
2. As we increase n_components across Dimensionality reduction, we see that the information loss is lesser, and the images become clearer. This can also be seen the calculation of MSE in the notebooks.
3. Experiments were carried out with KMeans and GMM as the results are interpretable, the algorithms are efficient, and it helps us understand the behaviour as per number of clusters. When experiment with algorithms like dbscan, these techniques fail to find clusters as clear as these two techniques indicating kmeans and gmm do a better job at capturing features inside the two datasets.
4. Factor Analysis and T-SNE were preferred in the last experiment. Factor analysis helps in reducing the dimensionality of data by identifying latent factors that explain the observed correlations among variables. It allows for representing a large number of variables with a smaller number of factors, making it easier to interpret and analyze the data. T-SNE is effective at capturing non-linear relationships in high-dimensional data.