Project Report
Machine Learning in Computational Biology

George Rouvalis

Contents

1 Introduction

For this unsupervised learning assignment five single-cell synthetic datasets were provided. Each dataset provides the gene expression profiles (200 Genes) of 200 cells. The goal was to develop a data analysis pipeline that accepts as input a dataset and implements the following analysis stages in a pipeline:

- Dimensionality reduction:
  - PCA
  - TSNE
  - UMAP
- Clustering (for each dimensionality reduction technique)
  - Gaussian Mixture Modeling with BIC (Bayesian information criterion)
- Visualization of the results

2 Prerequisites - Notes

- The code was developed on a Google Colab notebook(.ipynb). Feel free to open it using Jupyter as well
- The following packages were used to apply dimensionality reduction:
  - PCA: sklearn.decomposition.PCA
  - TSNE: sklearn.manifold.TSNE
  - UMAP: umap-learn
- For clustering (Gaussian Mixture Modelling), sklearn.mixture.GaussianMixture was used
- Place the datasets or ensure that they are in the same directory as the notebook file

3 PCA

PCA (Principal Component Analysis) is a dimensionality-reduction method that is commonly used to reduce the dimension of the feature space of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

Reducing the number of variables of a data set comes at the expense of accuracy, but the core idea in dimensionality reduction is to trade accuracy for simplicity. It combines the input variables in a specific way, that the "least important" variables can be dismissed while still retaining the most significant parts of all of the variables

To sum up, the essence of PCA is simple:
Reduce the number of variables of a data set, while preserving as much information as possible.

First step of PCA is the **standardization** of the data. The PCA calculates a new projection of the data set and the new axis are based on the standard deviation of the input variables. So a variable with a high standard deviation will have a higher weight for the calculation of axis than a variable with a low standard deviation. After normalizing the data, all variables have the same standard deviation, thus all variables have the same weight and PCA calculates relevant axis.

3.1 Number of Components

The second step is common along all dimensionality reduction methods of this assignment. That is to inspect for every different value of the target parameter the gradient of the BIC scores curve while applying Gaussian Mixture Modelling. Intuitively, the concept of the gradient is simple: if two consecutive points have the same value, their gradient is zero. If they have different values, their gradient can be either negative, if the second point has a lower value, or positive otherwise. The magnitude of the gradient tells how much the two values are different. Consequently, the idea is to look for the *"elbow"* in the BIC curve, or where the gradient stops decreasing.

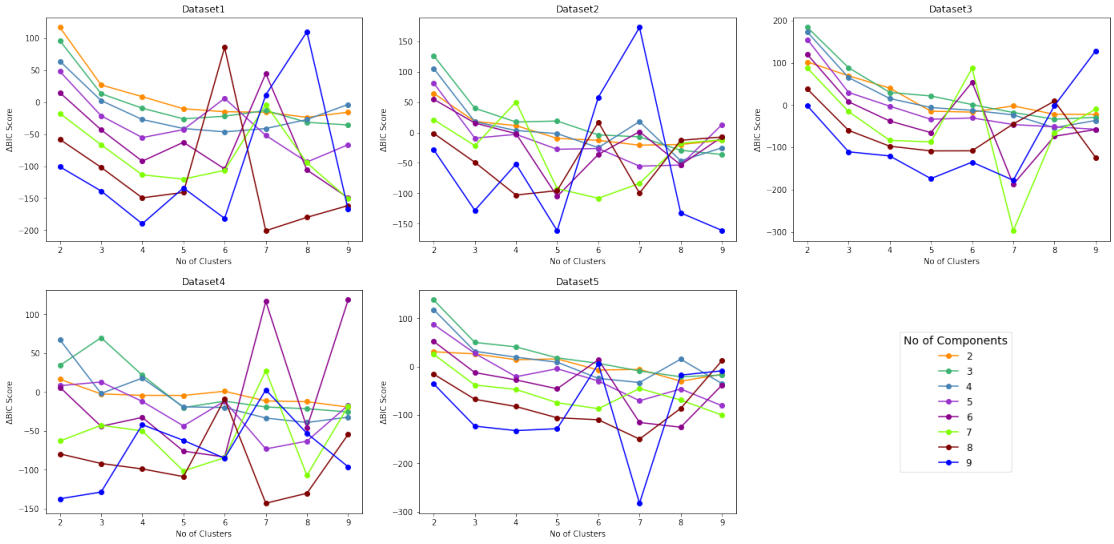

Figure 1: Component Parameter on ΔBIC per cluster number

Investigating each line plot from Figure 1, the smoother curve appears to be the one with the 4 components. Therefore, each dataset will be reduced to 4 principal components and will be fed to the GMM model.

3.2 GMM

Before applying clustering to all of the reduced datasets, the optimal number of clusters/states must be determined by calculating the BIC scores.

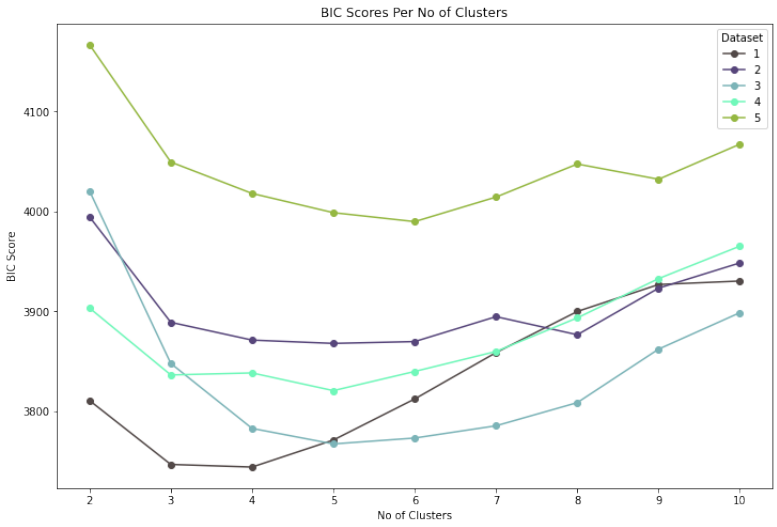

Figure 2: BIC scores using the optimal number of PCA components (=4)

The idea now is to find a reasonable number/local minimum where the most datasets present a low BIC score. The clustering process can then be initiated by using that number of clusters. These are the results of clustering on all the datasets:

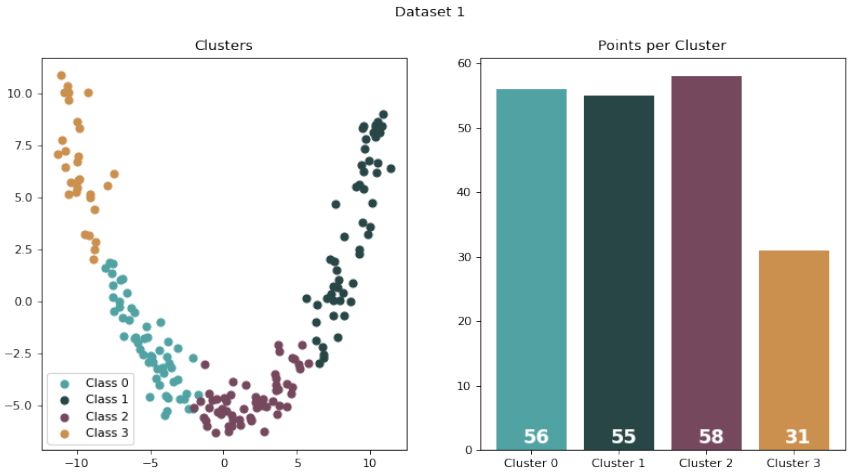

Figure 3: Dataset1 Clusters

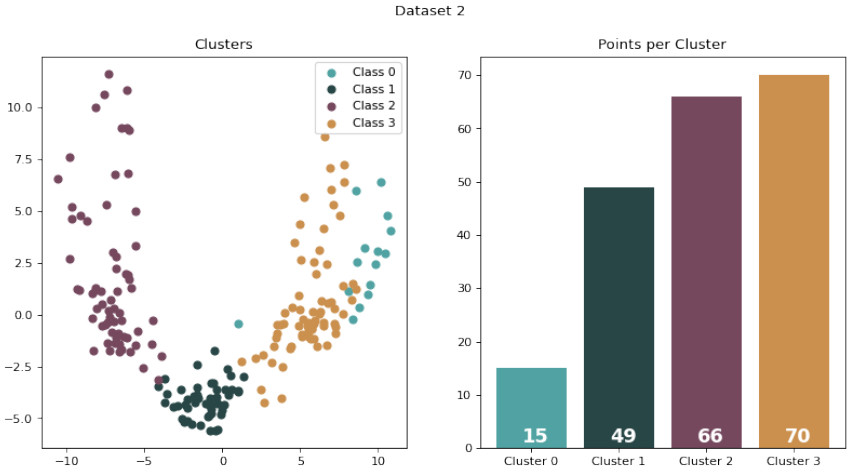

Figure 4: Dataset2 Clusters

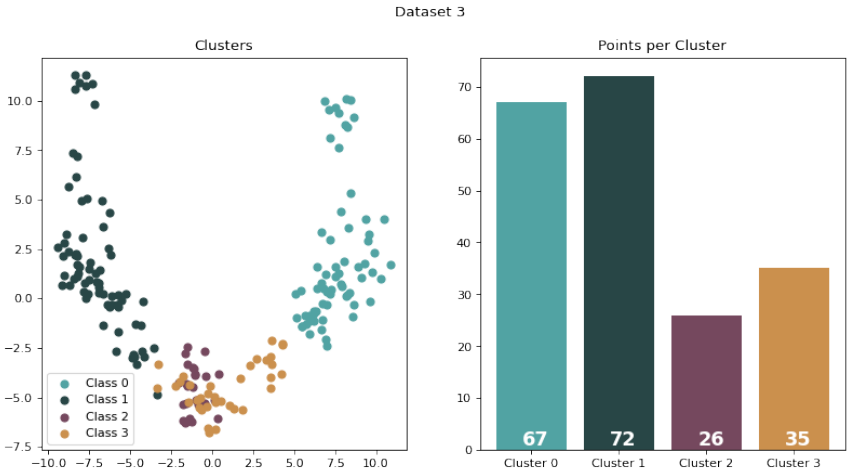

Figure 5: Dataset3 Clusters

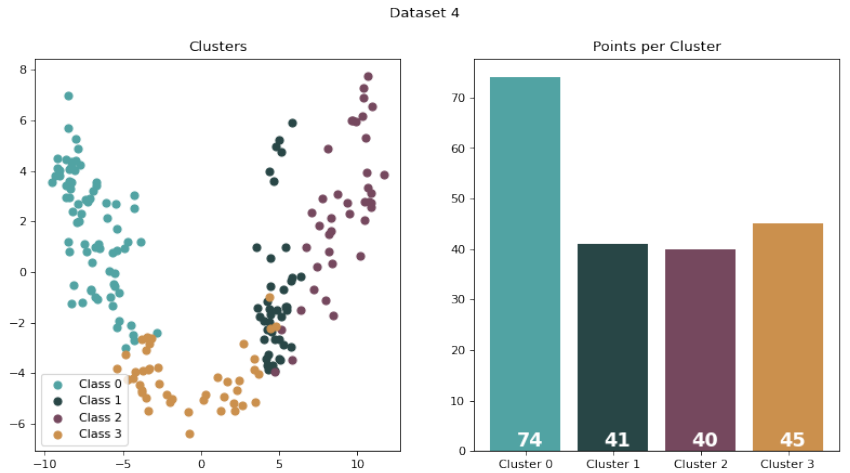

Figure 6: Dataset4 Clusters

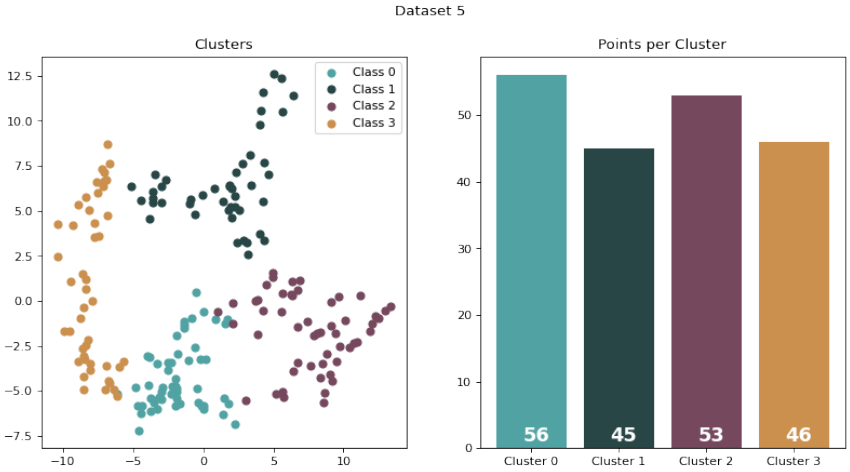

Figure 7: Dataset5 Clusters

Regarding PCA, if the most important variables are the ones that happen to have the most variation in them, then PCA is a credible method of feature selection. Unfortunately, when the components were selected automatically by the PCA library in order to fulfill the 95% explained variance, the features were reduced from 200 to ∼100, which is not a satisfying number at all.

In this case, by trying different smaller PCA component numbers, one can see the their effect on the GMM application and choose the appropriate one based on the previously discussed points.

4 TSNE

TSNE (t-Distributed Stochastic Neighbor Embedding) is an unsupervised, non-linear technique primarily used for for exploring high-dimensional data. In other words, TSNE gives you a feel or intuition of how the data is arranged in a high-dimensional space.

It converts similarities between data points to joint probabilities and tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data. In general, it is highly recommended to use another dimensionality reduction method before applying TSNE, but in in this project that doesn't take place as the comparison between the techniques is the main topic.

4.1 Hypertuning: *perplexity*

The first critical parameter available for hypertuning is called "perplexity," which represents (loosely) how to balance attention between local and global aspects of the data.The perplexity is related to the number of nearest neighbors that is used in other manifold learning algorithms. Larger datasets usually require a larger perplexity and different values can result in significantly different results. The original paper says, *"The performance of TSNE is fairly robust to changes in the perplexity, and typical values are between 5 and 50."*

For this reason, the effect of multiple values of perplexity on gradient BIC scores is tested:

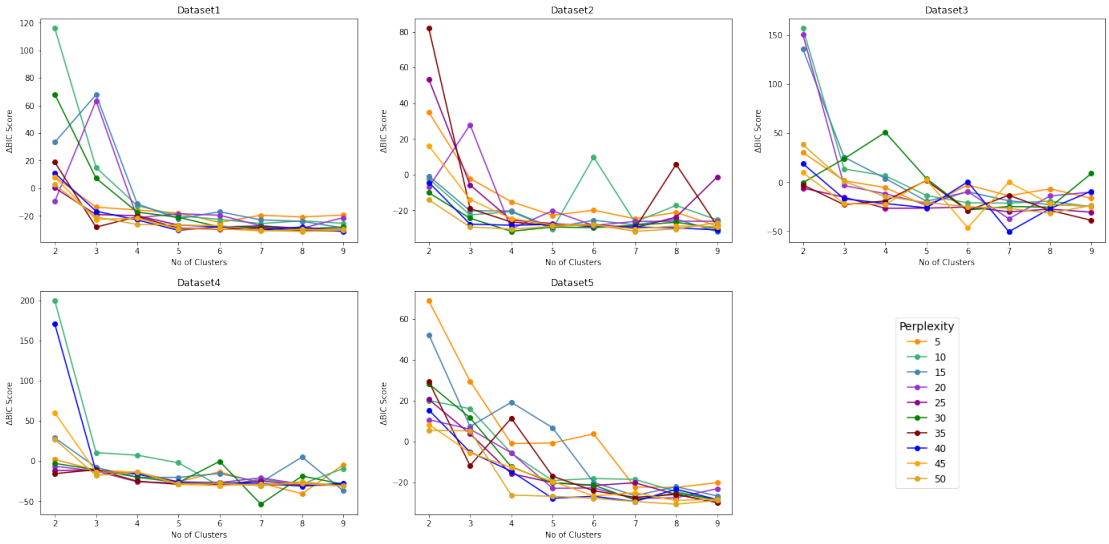

Figure 8: Perplexity Parameter on ΔBIC per cluster number

4.2 GMM

Observing the different plots for each dataset in Figure 8, the optimal value of perplexity seems to be 30 as it shows the least steep "slopes". Once again, this value is used for seeking the desired number of GMM clusters:

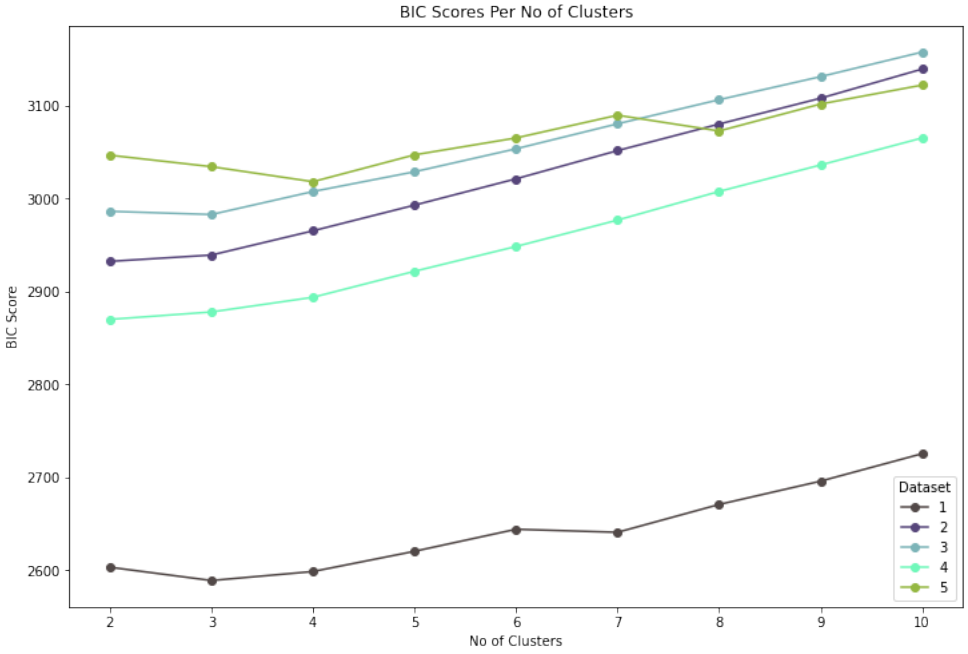

Figure 9: BIC scores using the optimal perplexity (=30)

Investigating Figure 9, a good choice for the number of clusters would be 3. The clustering procedure can start:

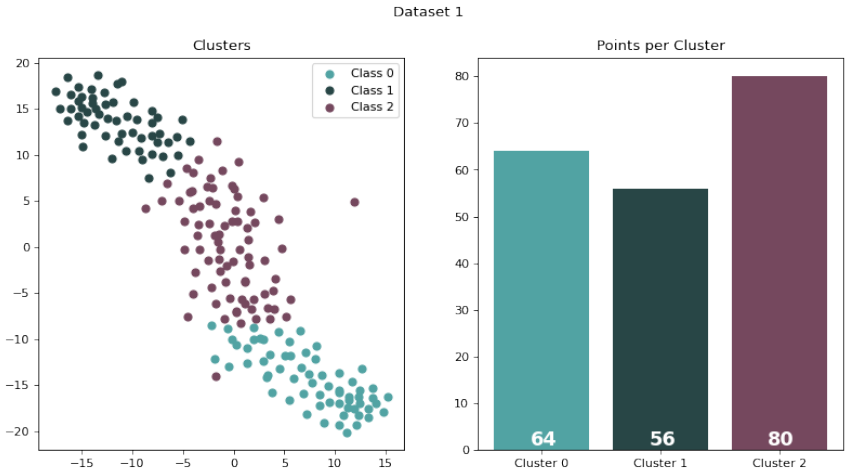

Figure 10: Dataset1 Clusters

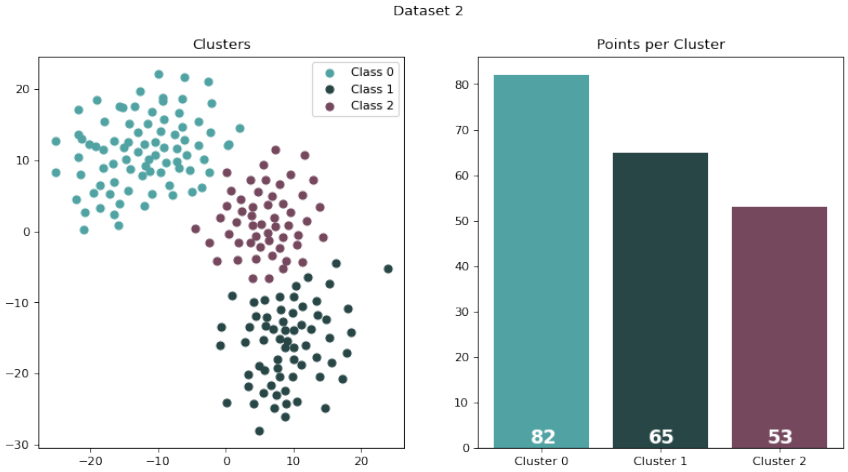

Figure 11: Dataset2 Clusters

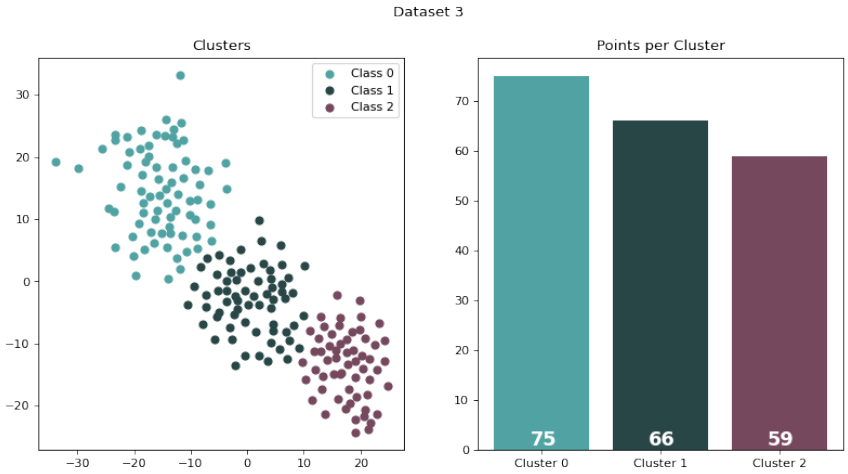

Figure 12: Dataset3 Clusters

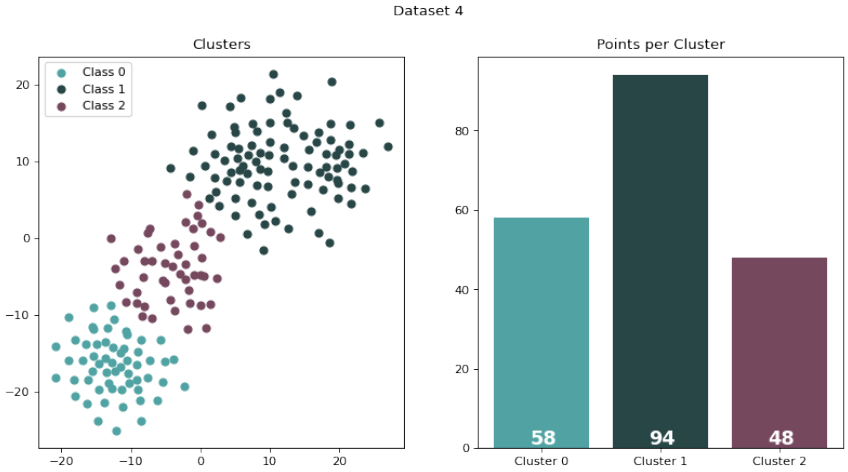

Figure 13: Dataset4 Clusters

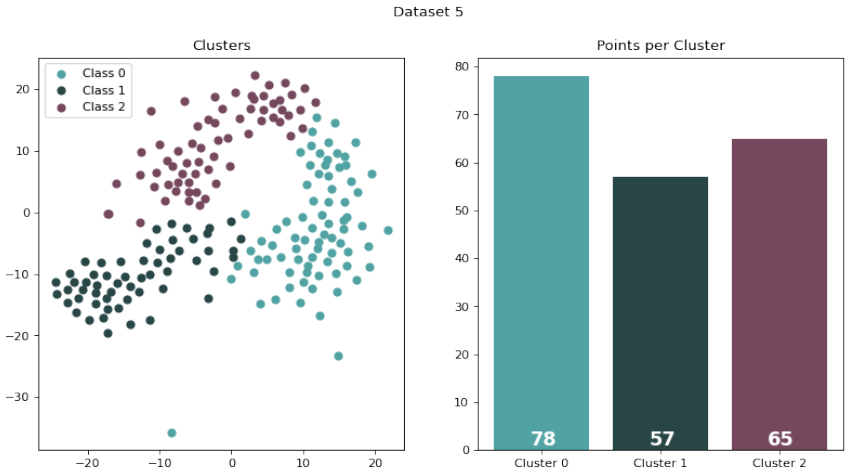

Figure 14: Dataset5 Clusters

5 UMAP

Uniform Manifold Approximation and Projection (UMAP) is a dimension reduction technique that can be used for visualisation similarly to TSNE, but also for general non-linear dimension reduction. The algorithm is founded on three assumptions about the data:

- The data is uniformly distributed on Riemannian manifold;
- The Riemannian metric is locally constant (or can be approximated as such);
- The manifold is locally connected.

From these assumptions it is possible to model the manifold with a fuzzy topological structure. The embedding is found by searching for a low dimensional projection of the data that has the closest possible equivalent fuzzy topological structure.

UMAP is very efficient at embedding large high dimensional datasets. In particular it scales well with both input dimension and embedding dimension.

5.1 Hypertuning: *n_neighbors*

The most critical hyperparameter is n_neighbors which represents the number of approximate nearest neighbors used to construct the initial high-dimensional graph. Essentially, It controls how UMAP balances local versus global structure. This means that low values of n_neighbors will force UMAP to concentrate on very local structure (potentially to the detriment of the big picture), while large values will push UMAP to look at larger neighborhoods of each point when estimating the manifold structure of the data, losing fine detail structure in trade of the broader picture of the data.

The effect of this hyperparameter on the gradient of BIC on the different datasets can be seen below:

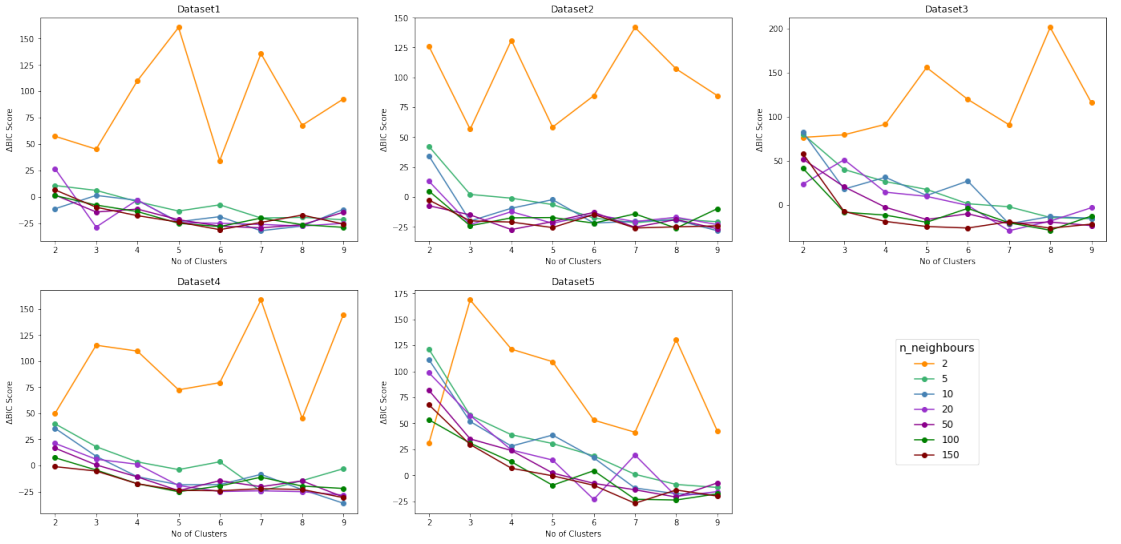

Figure 15: n_neighbors parameter on ΔBIC per cluster number

5.2 Hypertuning: *min_dist*

This parameter controls how tightly UMAP is allowed to pack points points together, with low values leading to more tightly packed embeddings. Lower values of min_dist will result in clumpier embeddings. Larger values of min_dist will prevent UMAP from packing points together and will focus on the preservation of the broad topological structure instead.

The default value for min_dist (as used on the previous hyperparameter tuning) is 0.1.

Once again, BIC scores seem to get significantly affected by the variations of this hyperparameter:

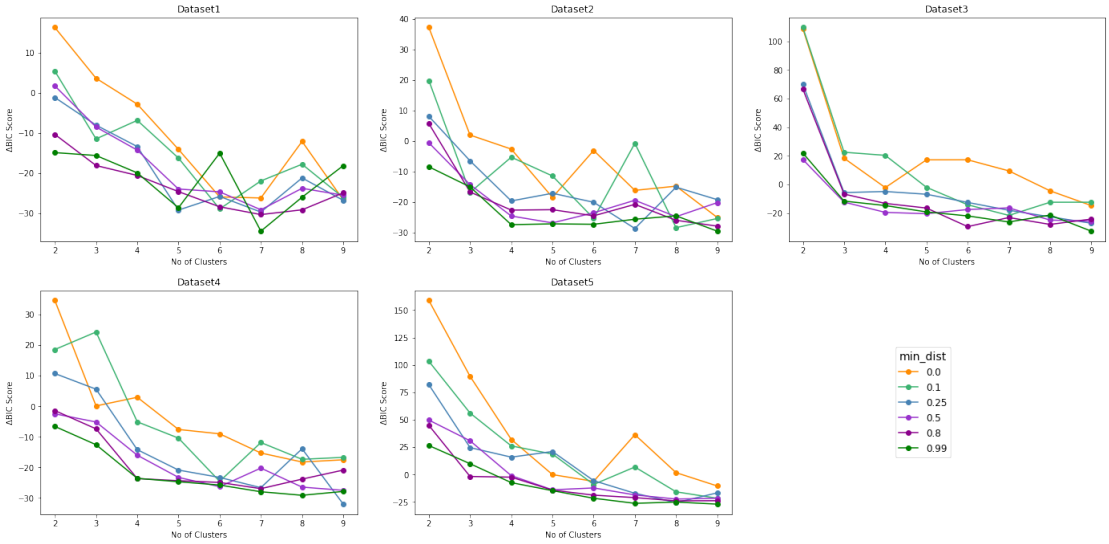

Figure 16: min_dist parameter on ΔBIC per cluster number

5.3 GMM

Noticing Figures 16 and 16, the least "unstable" plots derive from the values of n_neighbors=100 and min_dist=0.8. Time to calculate BIC scores based on these parameter values:

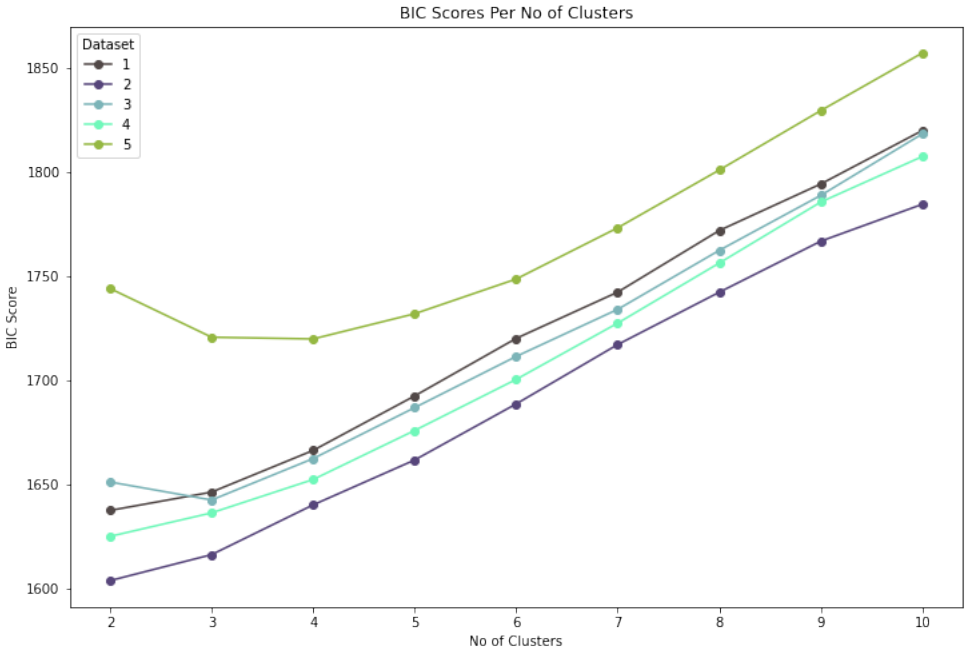

Figure 17: BIC scores using optimal n_neighbors (=100) and min_dist (=0.8)

The optimal number of clusters seems to be 2. Here are the results from GMM application:

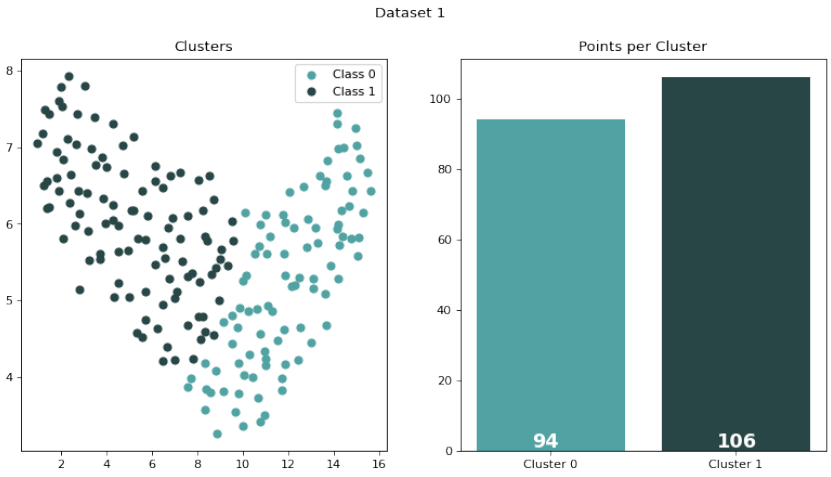

Figure 18: Dataset1 Clusters

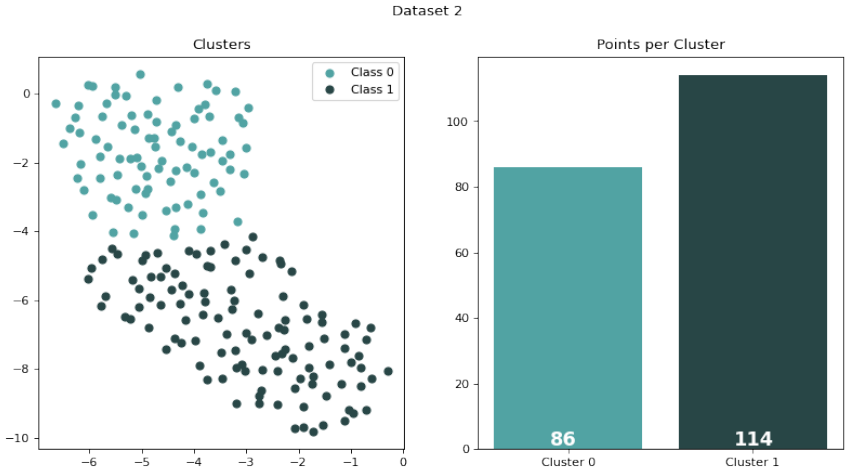

Figure 19: Dataset2 Clusters

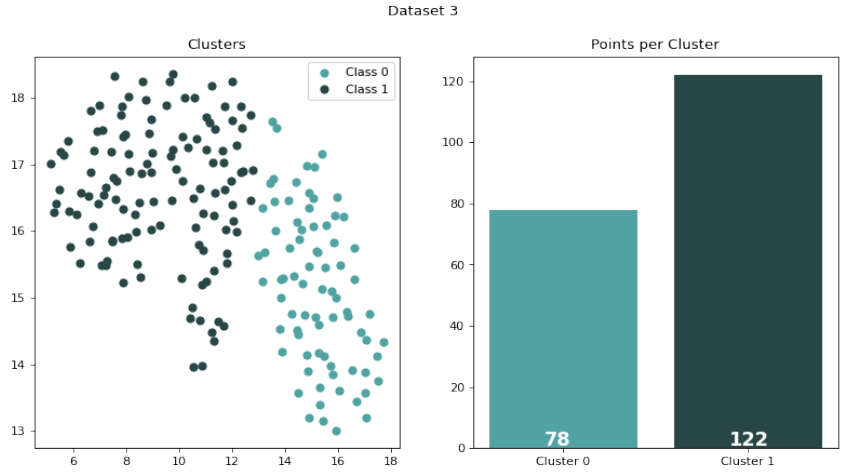

Figure 20: Dataset3 Clusters

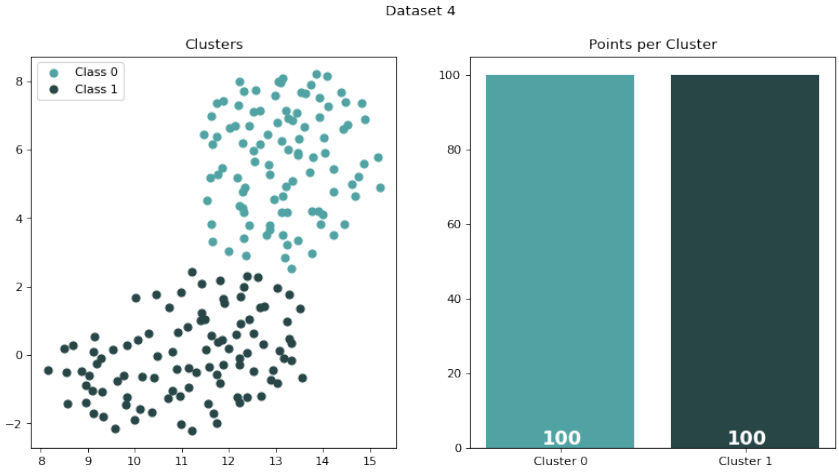

Figure 21: Dataset4 Clusters

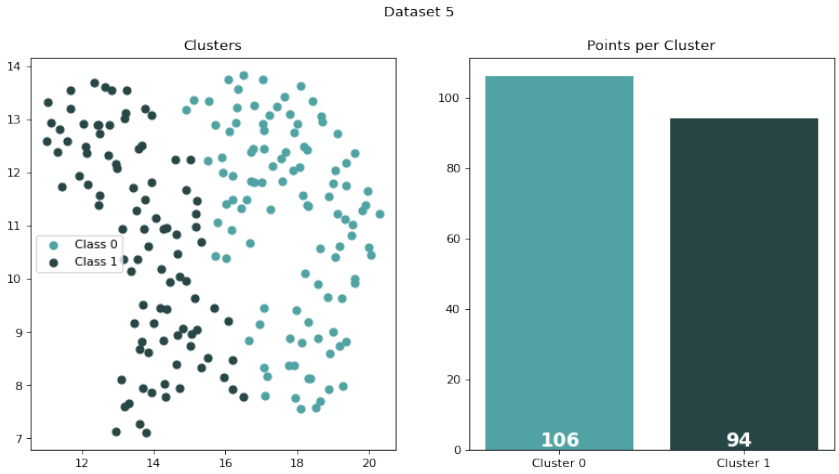

Figure 22: Dataset5 Clusters

6 Key Notes

The advantages and disadvantages for each dimensionality reduction method must be addressed in order to result to the ideal and generalized option for the optimal pipeline:

6.1 PCA

Pros:

- Correlated features are removed. There is no correlation between the principal components

Cons:

- Can detect only linear relationships between variables/features. If this is not true, it will not give sensible results.
- Gets highly affected by outliers.
- Information loss if Principal Components aren't selected with care

6.2 TSNE

Pros:

- Handles Non Linear Data Efficiently
- Preserves Local and Global Structure

Cons:

- Computationally Complex
- Non-deterministic with different results each time
- Too many hyper-parameters to be defined empirically (dataset-specific)

6.3 UMAP

Pros:

- Increased speed
- Better preservation of the data's global structure. The inter-cluster relations are potentially more meaningful than in TSNE
- More understandable hyperparameters

Cons:

- When projecting to lower dimensions, any given axis or distance in lower dimensions isn't directly interpretable in the way of techniques such as PCA

7 Discussion

After inspecting the produced results from the data analysis pipeline and weigh the advantages and disadvantages for each dimensionality reduction method, UMAP seems the most promising of reducing effectively the feature space of the datasets, as the average BIC scores of GMM is lower in comparison to the other two.

UMAP is a powerful tool in Data Science, and provides a great deal of advantages over TSNE and PCA. While both UMAP and TSNE produce somewhat similar output, the increased speed, better preservation of global structure, and more understandable parameters make UMAP a more effective tool for visualizing high dimensional data.

Ultimately, no dimensionality reduction technique is perfect. Data is forced to be fit into reduced dimensions - and UMAP is no exception. Nevertheless, by building up an intuitive understanding of how the algorithm works and understanding how to tune its parameters, we can more effectively use UMAP over the other two to visualize and understand large, high-dimensional datasets.