Homework 3. Unsupervised Learning; Fall 2023, OMSCS; Hossein Sahour, PhD

Background

In this paper, we investigate the application of clustering methods and dimensionality reduction (DR) algorithms for data preprocessing and employ these methods for training an artificial neural network. We used two datasets and implemented 6 algorithms on each dataset including two types of clustering algorithms namely K-means and expectation maximization (EM) and four dimensionality reduction algorithms namely principal component analysis (PCA), independent component analysis (ICA), random projection (RP), and Multi-dimensional Scaling (MDS). The article is structured in three major parts: the first part examines two clustering algorithms; the second section implements four dimensionality reduction techniques, the third section clusters the data after dimension reduction; the fourth and fifth sections utilizes the DR techniques and clustering methods and uses the output for neural network training.

Datasets

Two datasets were used. The first dataset, aimed at predicting students' dropout and academic success, encompasses 4424 observations with 36 features encompassing academic, demographic, and socio-economic aspects. The target includes classes of dropouts, enrolled, and graduates represented as 0, 1, and 2 respectively. This real-world dataset helps understand factors influencing academic outcomes and aiding decision-makers in supporting at-risk students. Its preprocessing involved checking for missing values, normalizing predictors, and encoding the target variable into three classes (dropout, enrolled, graduate) for a multiclass classification. The second dataset focuses on early-stage diabetes risk prediction, comprising 520 instances with 17 features, including symptoms and patient demographics, collected from the Sylhet Diabetes Hospital in Bangladesh. This dataset, valuable for early diabetes detection, was preprocessed similarly by checking for missing values, normalizing predictors, and converting the categorical target variable into a binary class (0 absence of disease and 1 presence of disease).

Section1) K-Means and Expectation Maximization (EM) Clustering algorithms

K -Means is a popular clustering algorithm for dividing a dataset into distinct, non-overlapping clusters based on similarity. This process starts with the random or heuristic selection of K centroids (where K is the predefined number of clusters). Data points are then assigned to the nearest centroid measured by Euclidean distance (in this exercise). The centroids are recomputed as the mean of all points in the cluster, and these steps are iteratively performed until the centroids stabilize, indicating convergence. Despite its computational efficiency and effectiveness with large datasets, K-Means assumes that clusters are spherical and of similar size, which might not be the case in complex real-world data scenarios. The algorithm's results can also vary based on the initial selection of centroids.

Expectation Maximization (EM) is a more advanced algorithm than K-Means and is widely used for clustering and parameter estimation in models with latent variables, such as Gaussian Mixture Models (GMM).EM consists of two main steps: the Expectation (E) step, where it estimates the likelihood of each point belonging to a particular gaussian distribution, and the Maximization (M) step, where it updates the model parameters based on these estimated likelihoods. The algorithm iteratively performs these steps until the model parameters converge. EM's flexibility allows it to identify clusters of various sizes and shapes and supports soft clustering, where points can belong to multiple clusters with varying

membership degrees. However, it's more computationally expensive than K-Means and is also sensitive to initial parameter selection, with a risk of converging to local maxima.

We used internal and external metrics to evaluate the clustering algorithms. The internal evaluation uses the internal information of the clustering process to evaluate the quality of the clusters without reference to external data. External evaluation involves comparing the clustering results to an externally known result. In our case, these are the target variables in our datasets that are used as ground truth. We used the elbow method and Silhouette score as internal metrics and homogeneity and completeness scores as the external metrics. For the GMM model, we used AIC value instead of the elbow method because the inertia_ attribute used for calculating WCSS in KMeans is not directly available in GMM.

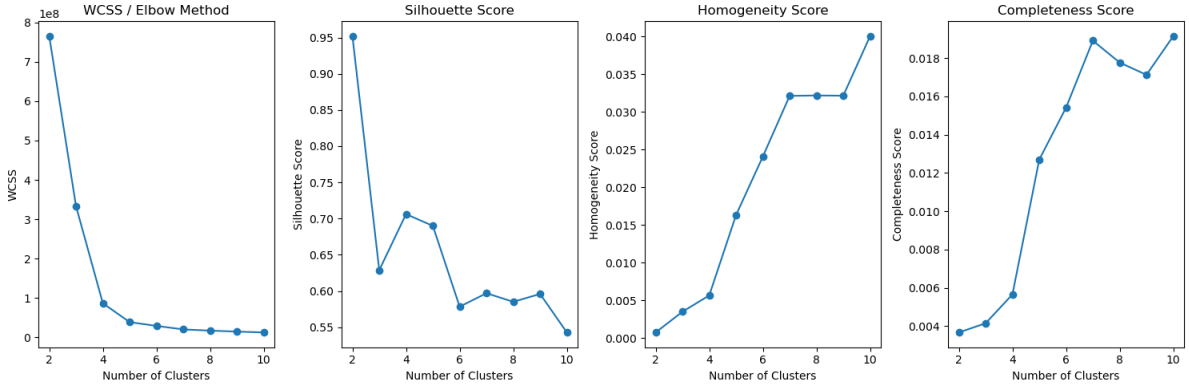

Figure1. The evaluation of K-means clustering on students' success and drop-off (dataset1)

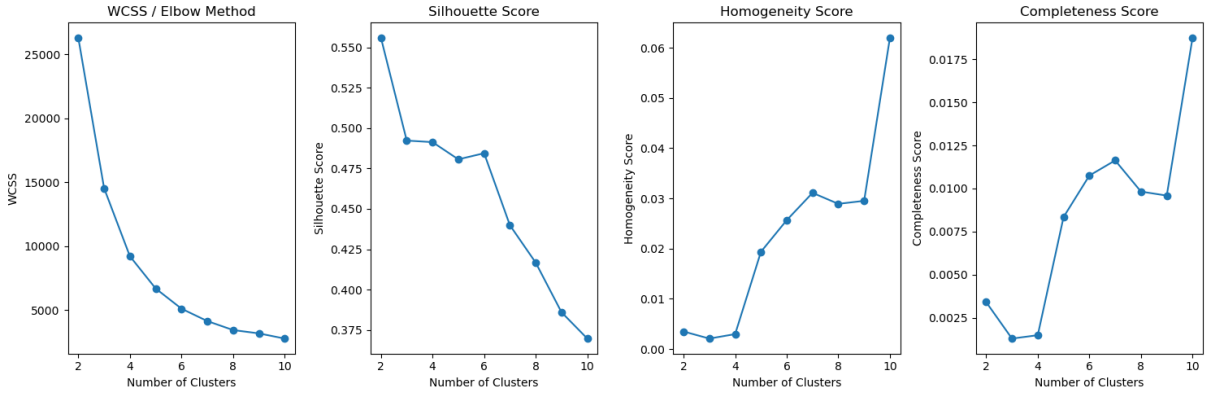

Figure2. The evaluation of K-means on early-stage diabetes risk prediction (dataset2)

Figures 1 and 2 show the evaluation of the K-means algorithms on dataset1 and dataset2. In Dataset 1 (Figure 1), which predicts student outcomes (dropouts, enrolled, graduates), the clustering analysis suggests an optimal number of clusters. The elbow method points to k=4. However, the silhouette score indicates optimal clustering at both k=2. This implies that, while four clusters distinguish the primary categories, two clusters might group students into broader segments (such as active vs. inactive students). Homogeneity and completeness scores improve significantly up to k=7, suggesting that finer subdivisions of students are possible and meaningful up to this point. However, the fluctuating scores beyond k=7 could indicate excessive segmentation. In general, 4 clusters can be considered an optimal

number of clusters when this division minimizes within-cluster variance while maximizing between-cluster differences, resulting in distinct, cohesive groups that are meaningful to group the students based on their socio-economic and academic status.

In the second dataset (Figure 2), aiming to predict early-stage diabetes using health factors, the "elbow" in the WCSS plot suggests 4 optimal clusters, while the Silhouette Score emphasizes a clear distinction at k=2, possibly differentiating the presence and absence of diabetes. The homogeneity score points to better-defined health profiles up to k=6, whereas the completeness score peaks notably at k=2 and then at k=10, indicating efficient grouping of health profiles. In essence, a 2-cluster solution might best distinguish diabetes risk but considering 4 clusters could also provide insights into varying risk levels.

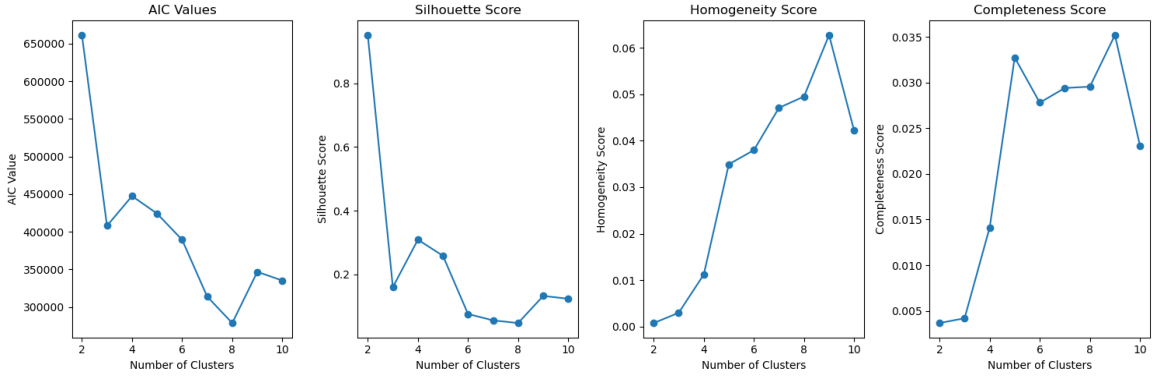

Figure 3. The evaluation of GMM clustering on students' success and drop-off (dataset1)

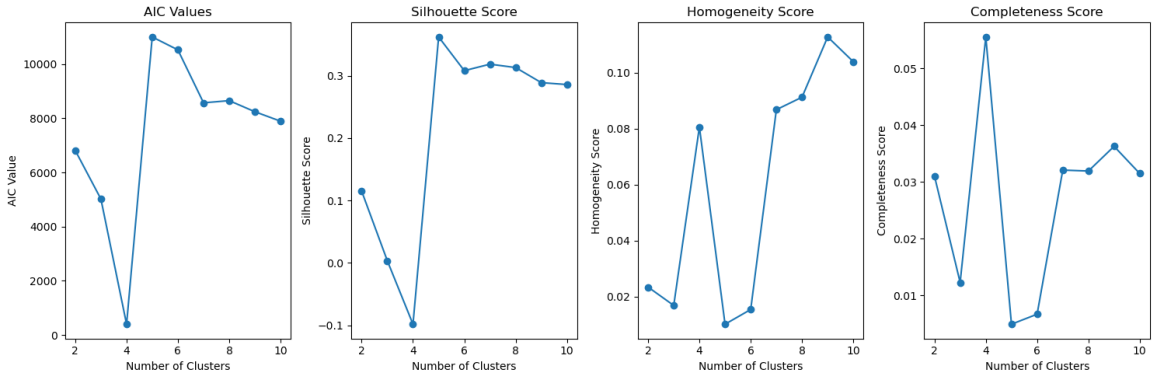

Figure 4. The evaluation of GMM on early-stage diabetes risk prediction (dataset2)

Analyzing the GMM results for the student dataset (Figure 3), the AIC values indicate a good model fit around 8 clusters where the curve almost stabilizes. The silhouette score confirms a strong distinction at k=2, perhaps separating students based on graduation likelihood. The continuous increase in homogeneity up to k=9 suggests more refined student groupings, with the completeness score also pointing to a detailed representation of student profiles at k=9.

For the diabetes dataset using GMM (Figure 4), the AIC values point towards an optimal fit at four clusters. The silhouette score identifies k=5 as the point of best data separation, perhaps categorizing patients based on different diabetes risk levels. Homogeneity peaks at both k=4 and k=9, indicating these points offer the most internally consistent groupings, while the completeness score suggests k=4 provides a full representation of health profiles.

The actual distribution, number of observations, and the dimensionality of features in dataset1, could be more aligned with the probabilistic modeling of GMM rather than the variance-based clustering of K-Means. The nature of dataset2 which includes patient symptoms may be more suitable for the probabilistic approach (GMM) than the variance-based approach of K-Means. This can be particularly true if the dataset contains overlapping clusters or if the clusters are not spherical, which is often the case in medical datasets.

Section2) Dimensionality reduction (DR) for each of the two datasets

Four DR algorithms used in this section include PCA, ICA, RP, and MDS. All four algorithms were applied to both datasets and the outcomes were visualized. PCA illustrates the cumulative explained variance by the principal components; ICA portrays the kurtosis of the independent components; RP showcases the reconstruction error as components vary; and finally, Multi-dimensional Scaling (MDS) plots the stress values for different numbers of components. Each technique's visualization provides insights into optimal component numbers or the nature of the data in the reduced dimensions for each dataset. The DR algorithms were set to start with all features (dataset-1: 36 features, dataset 2: 16 features).

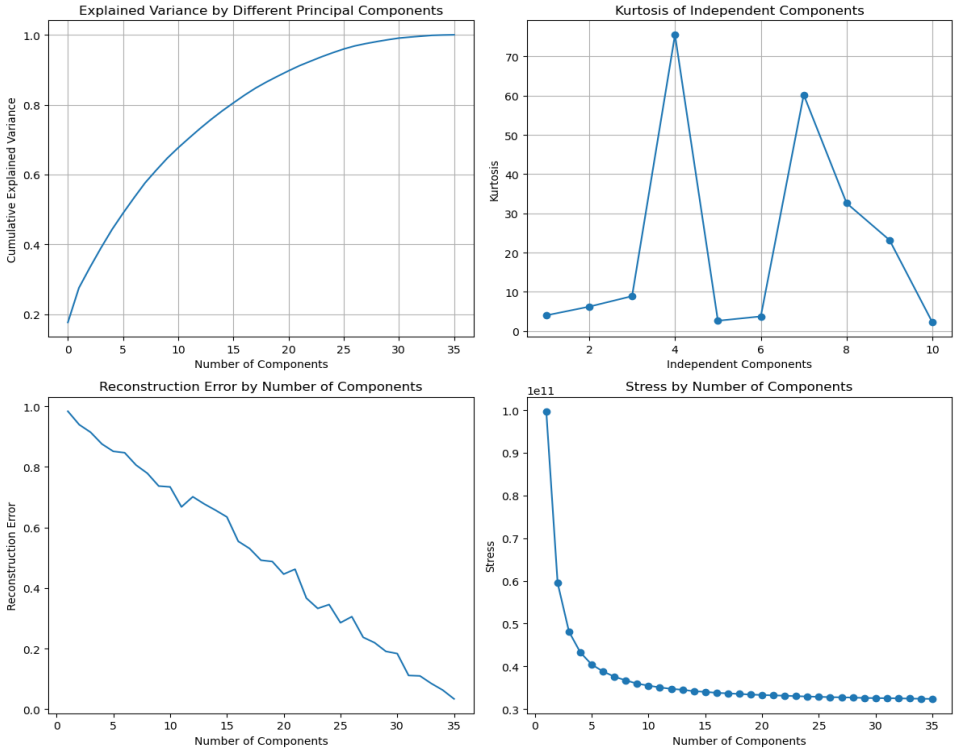

Figure 5 Dimensionality Reduction Analyses for dataset1 (predicting students' success)

Figures 5 and 6 show the results of the four DR algorithms on dataset1 and dataset2 respectively. According to Figure 5, the PCA plot reveals that a significant proportion of the variance in dataset 1 (with 36 features) can be explained by the initial components (around 25 components), indicating that dimensionality reduction might be beneficial without significant information loss. For the Kurtosis (ICA) we are looking for the highest values because it is less likely to be gaussian and more likely to be independent sources. Here the kurtosis plot for ICA suggests that certain independent components, especially around the 4th component, have a higher kurtosis value, indicating potential underlying

factors affecting students' outcomes. The reconstruction error from RP consistently declines as we increase the number of components, indicating more components might better represent the original data structure. However, the MDS plot indicates a sharp decline in stress for the initial components, which suggests that a lower-dimensional representation (around 25 as it stabilizes) can capture most of the pairwise distances in the dataset.

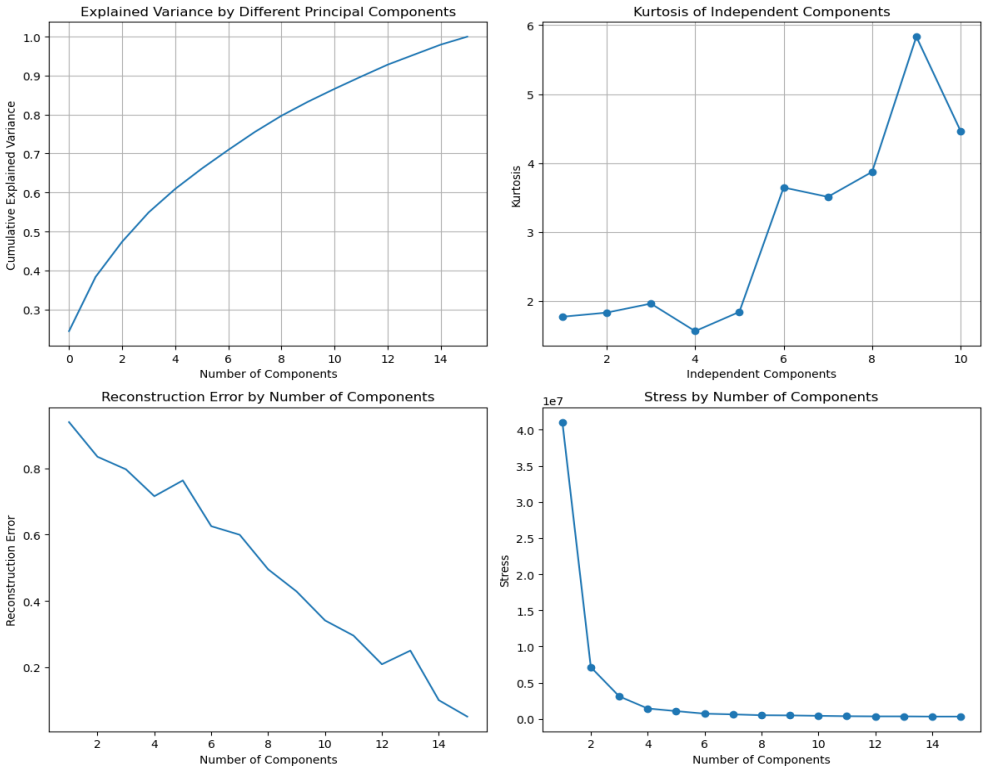

Figure 6 Dimensionality Reduction Analyses for dataset2 (predicting diabetes)

For Dataset 2, the PCA illustrates that about 12 components capture significant variance, suggesting an efficient data representation. The ICA's kurtosis demonstrates peaks at the 9th component, indicating distinct non-Gaussian structures. In RP, the reconstruction error steadily decreases as more components are added, hinting that more components retain better data integrity. The MDS indicates the possibility of achieving a quality two-dimensional representation of the data. These findings support utilizing a dimensionality reduction strategy that emphasizes the initial 12 dimensions for effectively predicting early-stage diabetes.

Section 3) Clustering algorithms with the output of dimensionality reduction

In this step, clustering algorithms are reapplied to the set of dimensionally reduced datasets providing 16 combinations of results. The results from PCA and K-means as well as MDS with K-means for each dataset are as follows.

The K-means clustering analysis for the MDS-reduced data displays varied results compared to the PCA output (Figures 7 and 8). The elbow graph for MDS shows a possible elbow at 4 clusters, although the decline continues beyond this point. The Silhouette Score in MDS peaks sharply at 2 clusters indicating optimal separation, in contrast to 3 clusters in the PCA results. The Homogeneity Score steadily rises with

the number of clusters, unlike the distinct peak seen with PCA. Additionally, the Completeness Score for MDS peaks at 3 clusters and then fluctuates. Both algorithms show mixed results with preferred number of clusters between 3 and 4.

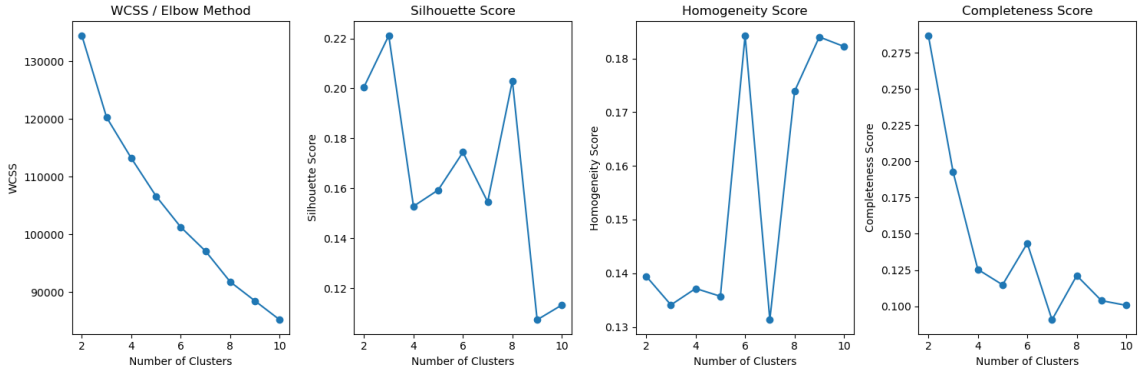

Figure 7. K-means with the outputs of PCA for dataset1

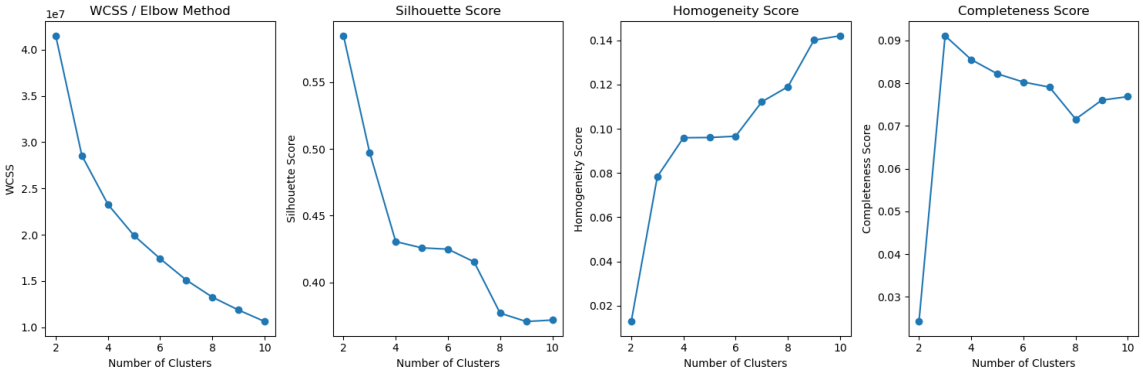

Figure 8. K-means with the outputs of MDS for dataset 1

According to Figures 9 and 10, MDS shows a preference for 5 clusters while PCA shows stronger preference for 4 clusters.

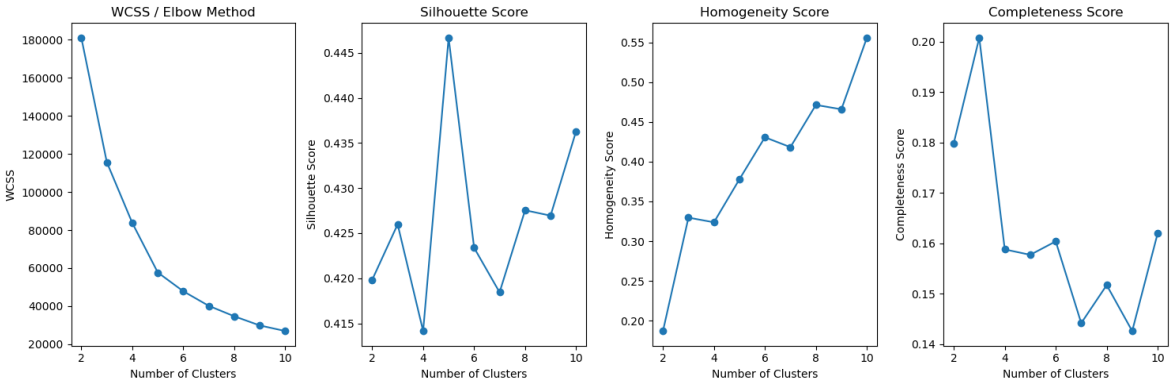

Figure 9. K-means with the outputs of PCA for dataset2

The outcomes of clustering from PCA and MDS shows some differences in patterns. PCA emphasizes the dimensions with the highest variability to maximize variance using orthogonal features. On the other

hand, MDS focuses on maintaining pairwise distances, uncovering possibly non-linear configurations and thus resulting in varying clustering results.

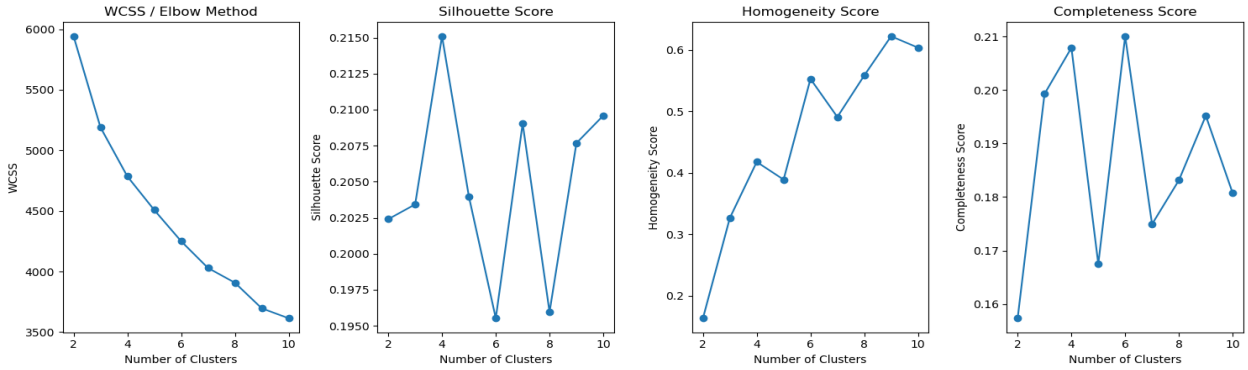

Figure 10. K-means with the outputs of MDS for dataset 2

Section 4) Fit an ANN to the new input space from the output of dimensionality reduction.

In this part, we selected the ICA and MDS algorithms to tackle dimensionality reduction. We used the dataset1 on academic success and failure. The reason for choosing this dataset is the presence of numerous and possibly overlapping features that make it suitable for dimensionality reduction. The results derived from these methods were subsequently applied to train an Artificial Neural Network (ANN). The configured architecture of the ANN comprises a hidden layer with sizes 32 and 16, employs the 'relu' activation function, and uses 'adam' as the solver.

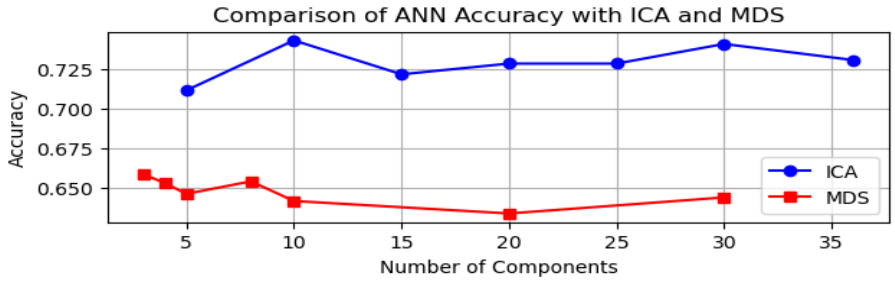

Figure 11. Accuracy of ANN`s prediction with different components

We plotted the accuracy of predictions with ANN for different components. ICA reached its maximum accuracy in 10 components and MDS in 3. The differences in the optimal number of components and subsequent accuracy of the fit can be attributed to the nature of the two algorithms. ICA determines statistically independent components within the data, which may need a larger number of components to capture independent sources of variation. MDS aims to preserve the pairwise distances between data points in a lower-dimensional space, which can be achieved with fewer dimensions.

The two ANN models (ANN with ICA and MDS outputs) were compared to the original ANN (without DR outputs) (Figure 12). Results showed a significant reduction in accuracy when MDS was applied to the dataset. The ICA-ANN and original ANN showed similar behavior in terms of accuracy and generalizability. This suggests ICA can reduce dimensionality without information loss. However, this did not enhance the prediction accuracy of the ANN.

Section 5) Fit an ANN to the output of clustering algorithms.

Kmeans and GMM clustering algorithms were applied to dataset 1(students success) with 4 clusters. These clusters were then treated as labels for the dataset, and used for subsequent predictive modeling with ANN. However, the clustering resulted in a modest overall accuracy of 0.52, which suggests potential room for improvement. One plausible explanation for this low accuracy can be the imbalance distribution of labels within the dataset. To enhance accuracy, it is recommended to apply techniques such as random oversampling to rectify the imbalance within the label distribution. Moreover, tuning hyperparameters and exploring the use of a different number of clusters can also be helpful.

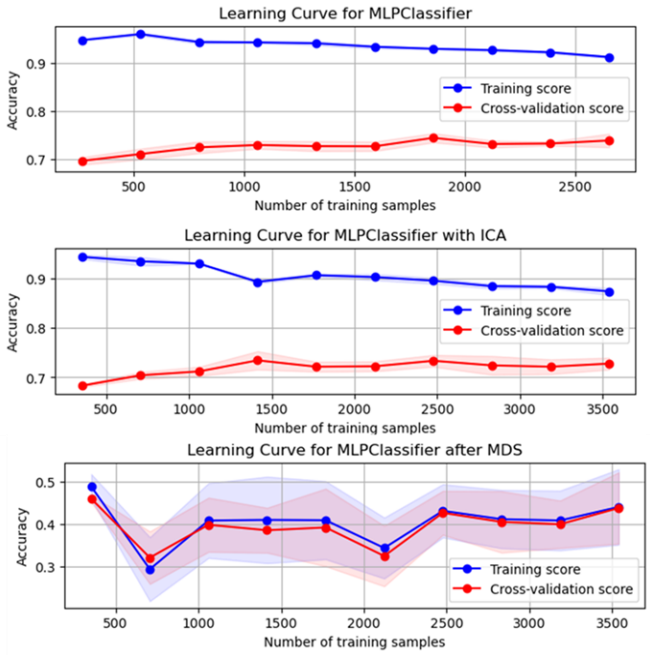

Figure 12 Learning curves for the three ANN models

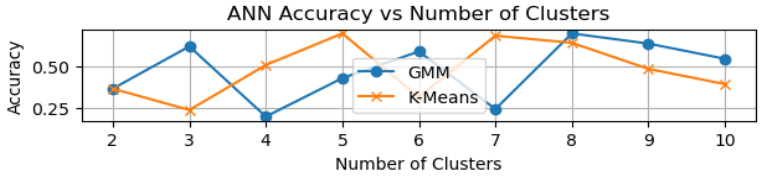

Figure 13. ANN accuracy with various number of clusters

Figure 13 shows the improvement in accuracy for ANN when classified into 5 and 8 clusters, comparing the performance between ANNs utilizing K-means-derived labels and those employing GMM labels.

Conclusion

The evaluation of clustering algorithms showed variable results for the two datasets. This emphasizes the role of domain knowledge as well as the purpose of clustering to come up with optimum number of clusters. The evaluation of DR algorithms demonstrated that while the optimum components for dataset1 and dataset2 were identified as 25 and 12, respectively, the subsequent application of an ANN suggested an alternative perspective. It was observed that utilizing only 10 components for dataset1 was sufficient to achieve high accuracy, underscoring the importance of iterative experimentation and validation in model selection.

References

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, pp.2825-2830.

Datasets: https://doi.org/10.24432/C5MC89 , https://doi.org/10.24432/C5VG8H