CS7641 - Assignment 3
Unsupervised Learning and Dimensionality Reduction

Ashish Panchal
OMSCS Student - College of Computing
Georgia Institute of Technology
apanchal33@gatech.edu

***Abstract*—This Assignment explores Unsupervised learning and dimensionality reduction techniques, where it aims to understand the similarities and differences between them. Additionally We also aim understand the impact of using them in supervised setting to engineer features over two different set of datasets, with inherent differences. We hypothesize changes brought in by such techniques result in different behaviour of prediction, where we validate this by contrasting our results in Assignment 1.**

***Index Terms*—Unsupervised learning, PCA, ICA, RP, IsoMap, KMeans, Expectation maximization, Neural Networks.**

I. Introduction

In our early life we frequency come across new and unseen objects. To make use of such objects we try to understand their structure and segregate them with their usefulness. While we come across new data, to make sense of it is beneficial to cluster them before exploring it additionally data as an entity can be enormous and very difficult to process, this is true to machine learning algorithms as well, which suffer from Curse of dimensionality, requiring records to the power of number of unknown parameters.

This Assignment aims to study 2 clustering techniques and 4 Dimensionality reduction techniques over 2 datasets, hand picked to create differences, and take a deeper dive into understanding how to optimize them, their properties and how to use them in the context of Neural Network Classification problem.

The following sections covers description of our initial Hypothesis, selected datasets, analysis for individual Clustering algorithms, Dimensionality Reduction algorithms, with dataset and parameter combinations. This report does not cover details of algorithm hyperparameter tuning and implementation and focuses only on the behaviour aspects.

II. Hypothesis and Data

Two datasets were choosen for this assignment as detailed below. Please note data description is provided for readers understanding is is similar to Assignment 1.

A. Task 1 : Wine Quality classification

The Wine Quality dataset, [1], provides psychochemical and sensor data for Portuguese "Vinho Verde" wine, consisting 6,463 records and 12 features post cleaning, of which 11 are continuous, and 1 (the "type") categorical with two categories.

The target variable, "quality," is ordinal, ranging from 3 to 9, with higher values indicating better wine quality. Notably, the dataset's quality distribution is imbalanced, with the majority falling into categories 5, 6, and 7 (32%, 46%, and 16% respectively), while the remaining four categories each represent less than 3% of the total distribution. To mitigate prediction errors on both extremes, a derived target feature "quality_cat is created," indicating "average or better" quality (quality > 5), resulting in two categories with a distribution of 36.7% and 63.3% for low and average-or-better quality, respectively.

Missing values rows accounted for only 0.5% of the total data were removed. To simplify analysis, both the "type" and "quality_cat" features were converted into Boolean variables, where "white" and "average or above" were assigned values of 1.

Univariate distribution of all features and the target variable, fig.(1) shows 8 out of the 11 numeric features exhibit significant skewness, except for "fixed acidity," "pH," and "alcohol." Different machine learning models may respond differently to these distributions, making it an interesting aspect to explore.

fig.(2) reveals no strong correlations in the data, except for "total sulfur dioxide" with "free sulfur dioxide" and the "type" feature, showing correlations of -0.72 and 0.7 respectively, these were not treated in the study. most of the features except alcohol show only weak correlation with target.

Lastly, the data was divided into an 80:20 ratio for the training and test sets. The training data was further used to create a cross-validation set, distinct from the actual training set.

B. Task 2: Gender classification

The Gender Classification dataset, sourced from Kaggle [2], contains synthetic data representing various facial attributes used for predicting gender.

This dataset comprises 5,001 records and 7 features. Notably, 5 of these features are categorical, while the remaining 2 ('forehead_width_cm' and 'forehead_height_cm') are continuous. The target variable can take two possible values: Male and Female, evenly split at 50% each. This balanced distribution differs from the dataset 1, which had slight imbalance.

With the exception of the 'long_hair' categorical feature, the distributions for other features are approximately symmetric, fig. (3). Additionally, both continuous variables exhibit balanced data in the lower 75%ile, with a slight skew toward the higher end. These features highlight a more balanced dataset, reducing potential model biases caused by skewness.

illustrates varying degrees of correlation between features, with no clear inverse relationships. Notably, there is a significant positive correlation among features related to the nose, lips, and gender with target. Similar to the dataset 1, data was split into training, testing, and validation sets.

This dataset offers an opportunity to examine how different machine learning models handle both categorical and continuous features. The balanced nature of the target variable and the variety of feature correlations make it an interesting case for model exploration and analysis.

C. Hypothesis Based on data

Based on the description of the data following are the initial hypothesis:

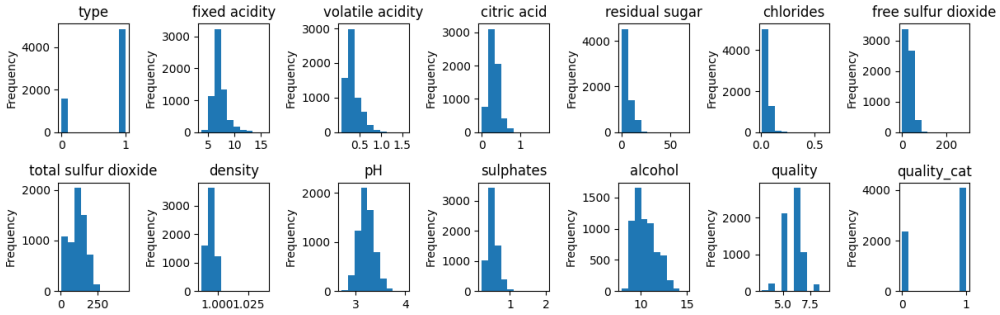


Fig. 1: Dataset 1 feature summary : univariate.

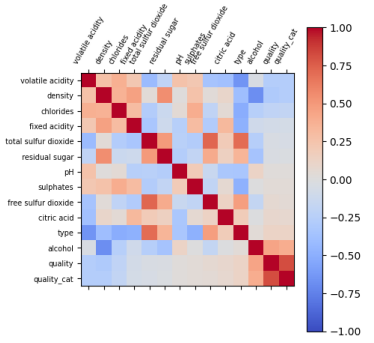


Fig. 2: Dataset 1 feature correla-

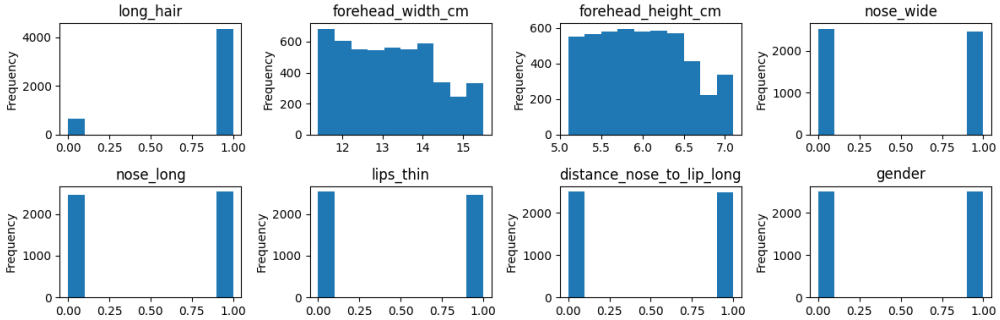


Fig. 3: Dataset 2 feature summary : univariate.

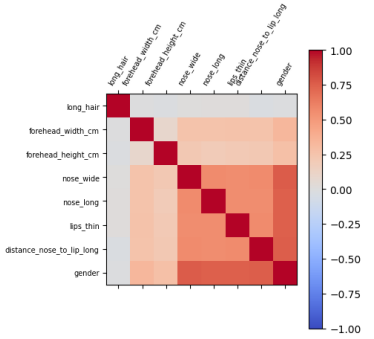


Fig. 4: Dataset 2 feature correlation summary.

1 Dataset 2 Due to its inherent segregation would result in low (close to two) best clusters.
2 Dataset 1 Due to its skewed nature will will difficult to cluster as it is, and therefore may required higher number of cluster to better represent the data.
3 Dataset 2 Due to its inherent segregation and more binary variables would result in low number of projections to represent the data.
4 Dataset 1 due to more continuous variables and conplex feature distributions would note result in significantly low projections to represent the data.
5 Dataset 2, Clustering with DR will not add much performance compared with raw scaled data
6 Dataset 1 would be better clustered with projections than raw scaled data.
7 Dataset 2, NN with DR with possibly reduced number of features would converge faster and with less overfit
8 Dataset 2, NN with Clustering will result in more sophisticated model, to capture new segregation in the data, and result in slightly longer training time however better performance.

III. Analysis of Each Algorithms Across Different Datasets

The following Sections covers as detailed analysis describing 5 part requirement of the assignment. Each section enclose a brief about the algorithms used for clustering or dimensionality reduction, over selected datasets, pertains relevant analysis and comparative insights.

IV. Clustering Techniques over Data : CTD

KMeans KM and Expectation maximization EM is implemented for both the datasets. The goal set for this exercise it to find the clusters which effectively segregate the data into clusters such that the intra-cluster distances are minimized at the same time inter-clusters distances are maximized, creating as much gap between clusters as possible. Silhouette Coefficient aligns well with this goal, for minimizing cohesion and maximizing separation. Where cohesion for a cluster is represented as avg intra cluster distance (a) and separation as minimum average distance from the nearest non membership cluster(b). Silhouette aims to maximize (b-a) while minimizing b. So the bigger Silhouette Score is better.

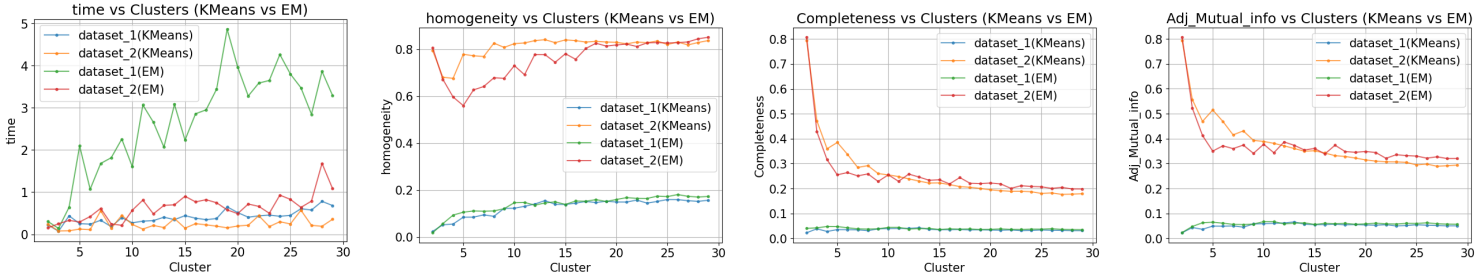
Assuming Gaussian noise in the data, Euclidean distance is used for both KM and EM, assuming maximum entropy in the data, although for dataset_1 unlike dataset_2 shows many features with skewed distribution. This might limit the performance of the distance unweighted by observed distributions, and a potential area of improvement.

A. *CTD : Kmeans*

KM was implemented and optimized to find the # clusters which maximizes the Silhouette score. From computational feasibility POV a range from 2 till 30 clusters were evaluated.

1) ***Dataset_1:*** 3 Clusters procured the Highest Silhouette score S' 0.23, *fig (6)*,slightly higher than 2 clusters, meaning prior ensures comparatively higher cohesion between clusters and separation. With the increase in the cluster count K, the S' decreases almost monotonically, i.e either the clusters are two close to each other or are sparse. This reflects the skewed nature of the features, which give smaller range to segregate points into distinct clusters. The peak S' is still very close to 0 pointing to low confidence in cluster assignment, further supporting the non-separability of data. This is not only true for Cluster features but also for the target labels y, where homogeneity (H) is overall close to 0, with a slight positive gradient,*fig (5b)*, reflects the clusters do not pertain to any single label class, which is also seen through Completeness score (C) and Adjusted mutual information (AMI) *fig (5c,5d)*, where the later points that the clusters are not very distinguishable to provide information about the other cluster, and are possibly equally random collection of features values. Lastly S' agrees with Elbow curve at 3, i.e the sharpest dip in the SSR, however the overall SSR is still very high.

2) ***Dataset_2:*** 3 Clusters Procures the peak S' 0.35,*fig (6)*, however with increase in the K, S' decrease and again starts



(a) Time vs Clusters (b) Homogeneity vs Clusters (c) Completeness vs Clusters (d) Adj Mutual info vs Clusters

Fig. 5: KMeans vs EM performance comparison over Dataset 1 and 2

increase, pointing towards possible local clusters, which is further confirmed by very H at higher K, however Higher K results in the low C, i.e since there are more clusters than the number of y, clusters are not exclusive which also results in ambiguity in dependence of clusters in one another, increasing similar clusters and decreasing AMI. Therefore the peak H captures a place with comparatively high H, C and AMI, but not the highest of K=2.

B. *CTD : Expectation Maximization*

1) ***Dataset_1:*** Procures the highest observed S' 0.42 at K=2,*fig (6)*, i,e these clusters while maximizing the likelihood of the centroids generating the data, assuming normal noise, also maximizes the cohesion ad separation between them. However, similar to Kmeans, increase in number of clusters drastically decrease these properties, doing much worse than Kmeans. Although the quality of clusters from H, C and AMI point of view is similar, following the arguments as previously provided. The model does not align well with highly skewed distribution of features, resulting in very close points, with increase in number of clusters, for EM, this might identify mean of the gaussians very close to each other resulting in very low S', which aims to maximize the distance between clusters, creating a disagreement between the algorithm and goodness measure defining the nature of the data X. In contrast the BIC, at larger K is lower, which also aligns with H, C and AMI, which are able to better explain the variance in the X, however the variation over K in BIC is low, pointing towards high unexplainable noise in the X.

2) ***Dataset_2:*** EM reaches it peak S' 0.345 at K= 2,*fig (6)*, comparable to Kmeans K=2 and slightly lower than its K=3. Post which S' slightly drops but only show fluctuations with change in K, i.e EM is able to find Gaussians with comparable performance at different K, this reflects of synthetic nature of the data with additive gaussian noise, and so gaussian means can be identified to explicitly explain the features. At peak S', unlike KMeans, H, C and AMI are maximum, aligning with the inherent 2 class segregation, 2 K could align with the labels, providing maximum mutual information with least similar clusters. The Goodness of fit measure align with the data and also with the algorithm. With such inherent qualities of data EM ensures segregation which explains it the most. In contrast to S', high K shows drastically low BIC, compared to Dataset1, i.e more number of Gaussians could explain the variance in X better.

3) ***Fit Time Comparison:*** Kmeans for both datasets and EM for dataset 2 show similar fit time over K, with a positive gradient. i.e more K results in more number of computations to segregate points in Kmeans KM and more number of probabilities to account for EM. However for dataset 1, EM takes 4x more time to find the ML mean of gaussians, this further supports the difficulty in segregating the clusters, due to skewed and noisy data. And could be indicative even at lower K that EM may not procure good results.

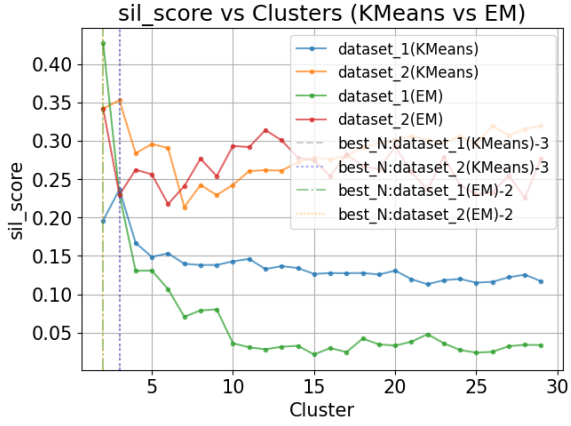


Fig. 6: KMeans vs EM performance comparison over Dataset 1 and 2 : Silhouette score

V. Dimensionality Reduction over Data : DR

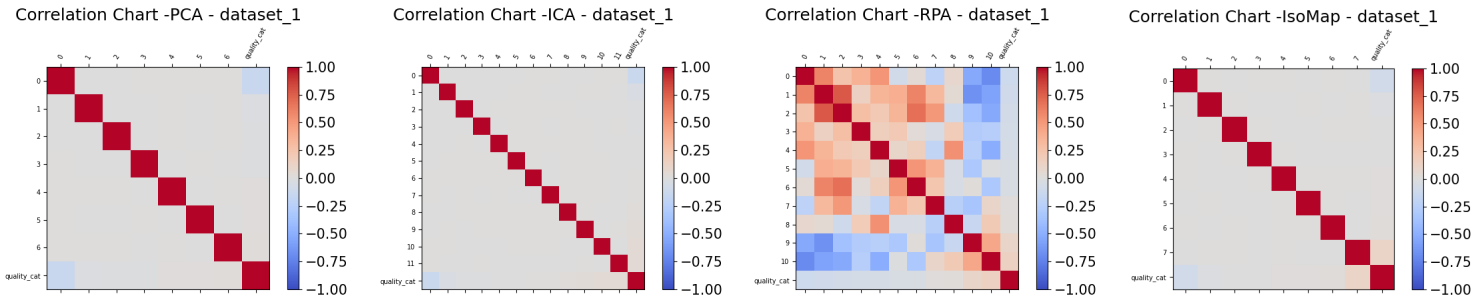
4 DR techniques are used for both the datasets to generate 4x2 projections, these are furthers used in the next sections for analysis. For Each projection dataset combination, the DR algorithm is optimized using as goodness using respective goodness of fit metric.

A. *DR : Principle component Analysis PCA*

Aim to identify orthogonal projections of the raw features which explain the maximum variance in the data, with an assumption that these projections which explain the higher variance could reconstruct the same data by comparatively less number of features, which tackles the curse of dimensionality. These projection can be identified as eigen vectors of the X where respective Eigen values (Ev) measures the magnitude of explained variance of the X. To examine the hypothesis in next section, least Number of highest eigen value principle components which could explain atleast 85% of the cumulative variance were selected.

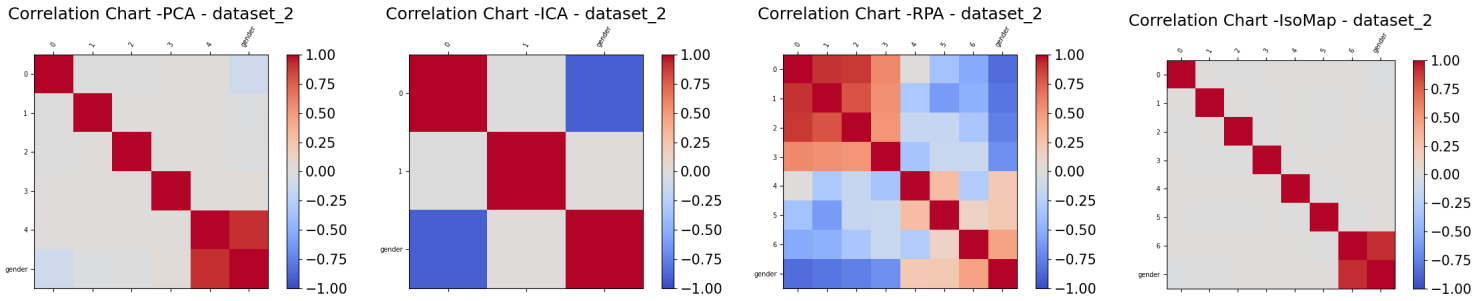
1) ***Dataset_1:*** The distribution of Ev is smooth,*fig (9a)*, i.e most projections still have significant explained variance compared to its lower and higher dimension. This is visible in Cumulative Explained Variance (CEv) as well, where last 2 projects have very small contribution. 7 features explain 86% of the CEv selected further for testing. Correlation chart of these projection shows no associated of features among each other as expected, however, only weak -ve correlation of one component with Y, signifying variance which does not explain Y in X,*fig (7a)*

2) ***Dataset_2:*** The Ev Distribution shows large dip from 1st component to 2nd,*fig (9b)*, similarly 4th to 5th, pointing towards possible segregation which explain the variance in the data well. Additionally even the lowest Ev component is significant 40. 5 components explaining 87.5% of CEv were selected. From the Corr. Chart one of the features is highly Correlated with Y, possibly with largest Ev.*fig (8a)*



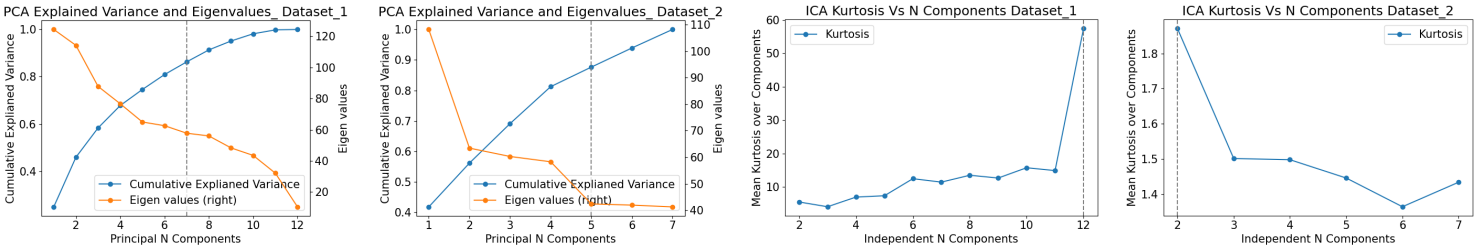
(a) PCA (b) ICA (c) RPA (d) IsoMap

Fig. 7: Dataset 1 Dimensionality Reduction Correlation charts



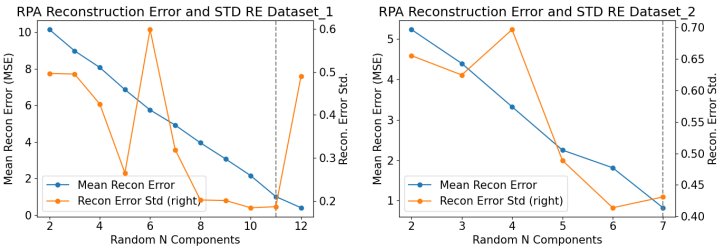
(a) PCA (b) ICA (c) RPA (d) IsoMap

Fig. 8: Dataset 2 Dimensionality Reduction Correlation charts



(a) Dataset 1 PCA (b) Dataset 2 PCA (c) Dataset 1 ICA (d) Dataset 2 ICA

Fig. 9: DR- Performance Comparison : (a,b)PCA-Explained Variance and Eigenvalues, (c,d)ICA-Kurtosis Vs N Components



(a) Dataset 1 (b) Dataset 2

Fig. 10: DR- Performance Comparison : RPA-Mean and STD Reconstruction Error

B. DR : Independent Component Analysis ICA

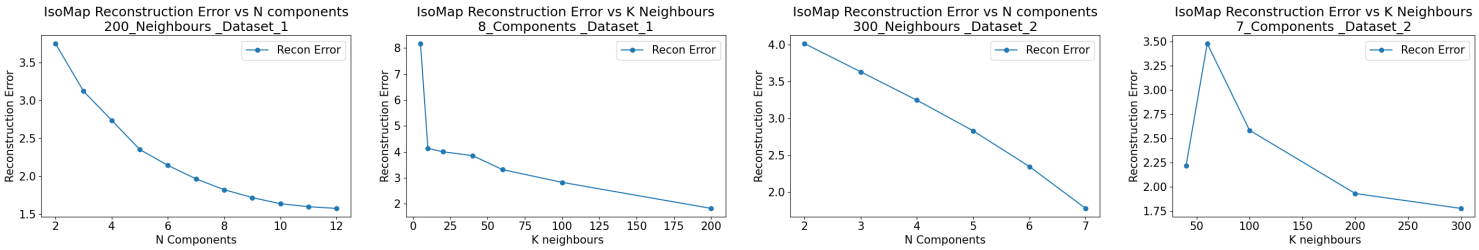
Unlike PCA, it assumes there are independent causal components linearly combined to create the observed Data X. Central limit Theorem (CLT) results In gaussian distribution on linear combination of independent features in limit. Therefore one way to identify independent components is with high Kurtosis signifying low non-normality. To evaluate a combination of N components, Avg kurtosis over them is calculated.

*1) **Dataset_1:** fig (9c)* shows, kurtosis 58, peaks at 12 components. Which is same as features in X, It is a difficult to say that causal independent components (IC) are more than 12 as they cannot be well segregated by 12 features. However it does mean that there are ¿= 12 IC, if exists, could explain the high variance in the features. However X contains features, assumed as combination of iid ICs, the combination of observed features should also create gaussian in limit, however most features in X are highly skewed and therefore it is unclear that that this assumption of ICA will still be beneficial. Similar to PCA, the fig shows no significant corr among features and only week -ve correlation between one of the features with Y, pointing towards, non-association of IC with Y additionally resulting of orthogonality in IC, creating same features as PCA,*fig (7b)*

*2) **Dataset_2:** fig (9d)*, Kurtosis 1.87 peaks at 2 components, signifying possibly 2 IC generating the data. This could follow from the synthetic gaussian noise added to X depending on Y. Therefore this inherent quality of data can be exploited by ICA, significantly reducing curve of dimensionality. *fig (8b)*

C. DR : Random Projections RPA

Creates Random Projections of X over N components. It assumes that the distances between higher dimensional vectors can be preserved to a degree when projected in lower dimension space, therefore on average reducing N. For selection of N components Reconstruction error (RE) was used, averaged over 5 random instances. That is the pairwise average square difference between the inverse transform of Projections and the original data. Where the worst was always with minimum components (2). Least N which reduced max RE by at least 80% was selected.



(a) Dataset 1 N Components (b) Dataset 1 K Neighbours (c) Dataset 2 N Components (d) Dataset 2 K Neighbours

Fig. 11: DR- Performance Comparison : IsoMap-Reconstruction Error : N Components and K Neighbours

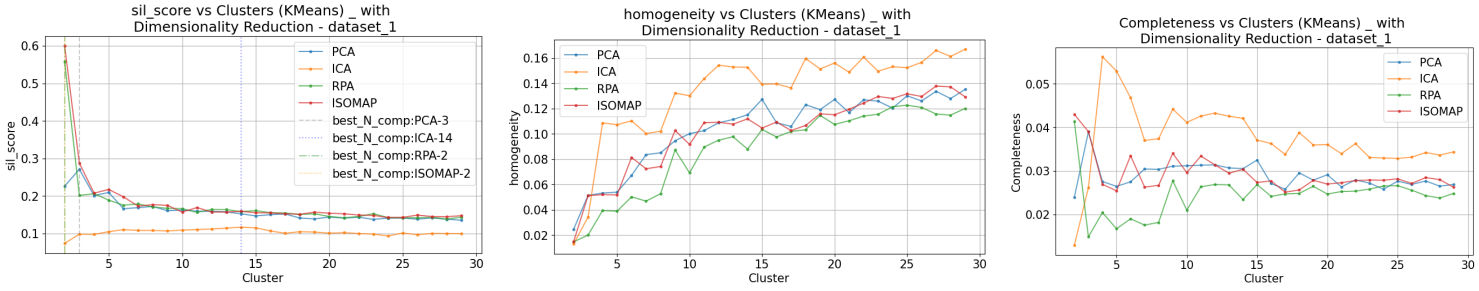
1) ***Dataset_1****: fig (10a)*,N = 11 with RE 1, reduced by 90% from max RE, with 0.18 STD of RE. Does not drastically decrease N, this could be supported by PCA where each principle component adds significant explainable variance, RPA component being combinations of PCA combinations also have the same properties. Therefore more combinations would be required in order to explain the variance. Hence less components may result in higher RE. STD mostly decreases with RE. Fig, since Random combinations contain some aspect of the single important Component identified in PCA, more components in RPA become correlated with target.*fig (7c)*

2) ***Dataset_2****: fig (10b)*,N = 7 with RE 0.82, reduced by 85% of max RE, with 0.43 STD. N components here are same as original components. Like PCA most components showed significant Ev, which would also follow in linear combinations. These combinations dilutes the CEv therefore more components are required to capture it significantly. Like Dataset 1, the projections are more correlated with the target as compared to X. *fig (8c)*

D. ***DR : IsoMap***

Unlike other methods creates a projection of the data onto a nonlinear space, while reducing the geodesic distance between instance pairs, and maximizing explained variance by identifying eigen vectors in this space. Since the nearest neighbour to calculate the geodesic distance is important, both K neighbours and N components were turned to find the lowest N and smallest K which reduces the RE from Max RE by at least 80%, if none then use the lowest RE N and K. Compared to other methods it is more intricate to analyze, therefore the below covers a brief about it under the scope of this assignment. Charts provided are variance in performance by keeping all but one property constant from the best model found.

1) ***Dataset_1****: fig (11a,11b)*,N=8 and K=200, RE 1.82 reducing over max RE by 85%. With Large number of neighbours K data show the RE reduces with increase in N components, this gradual change depicts the more # of dimensions could bring similar instances closer, however very large number of dimension may not bring the instances closer than the next lower dim. For fixed 8 Components, The RE reduces drastically at smaller K, therefore, reducing overfit. Although with further increase in K RE decreases more slowly, it is hypothesised that it may increase at very large K, due to direct connections to non-similar instances as well, while it was not explored due to computational limitations. Additionally, it is of concern that with further increase K if RE continues to increase in limit, then all the instances are equally random in nature. The observed results may depict that either there is a higher order non-linear relationship or IsoMap resulted in model overfit. *fig (7d)* shows only week correlation of 2 projections P with Target, similar to PCA and ICA. That i.e P which being similar instances closer over non-linear surface do not show strong association with Target variable. Therefore, this is a resultant of overfit to noise in X.



(a) Silhouette Score Vs K (b) Homogeneity Vs K (c) Completeness Vs K

Fig. 12: Kmeans -Dataset 1 : Performance Measures with Dimensionality Reduction

(a) Adj Mutual Info vs K (b) Squared error Vs K (c) Fit Time Vs K

Fig. 13: Kmeans - Dataset 1 : Performance Measures with Dimensionality Reduction

*2) **Dataset_2**: fig (11c,11d)*,N = 7 and K=300, RE 1.77 reducing over max RE by 72%. In this case the threshold was not reached therefore the smallest RE model was picked, this signifies either the worst model was not much worse in segregation of similar instances or the features have very large variance to be captured by 7 components. The model shows a monotonically decreasing trend in RE over N, like Dataset 1. However, with 7 components the RE spikes at 60 K, then continues to decrease, which signify that there are local similarities and global similarities, however there could be adjacent regions of non-similar instances. Therefore it can be expected that IsoMap with larger K captures these global similarities. Like PCA, *fig (8d)*, shows, high correlation between only one features with the target, i.e Target is possibly one of the major projection in identifying the similar instances.

VI. Clustering Techniques over Reduced Data - KMeans: CTRD

From Sec IV-B, shows low performance of EM on dataset 1 due to non alignment of the nature of data with goodness of fit metric and algorithmic assumption, it is found to be slow as well. Therefore Kmeans is explored to further study the nature of data. Additionally section 2.2, ICA which aims to find the IC of X, will be interesting to study the changes in method of segregation.

A. Dataset 1

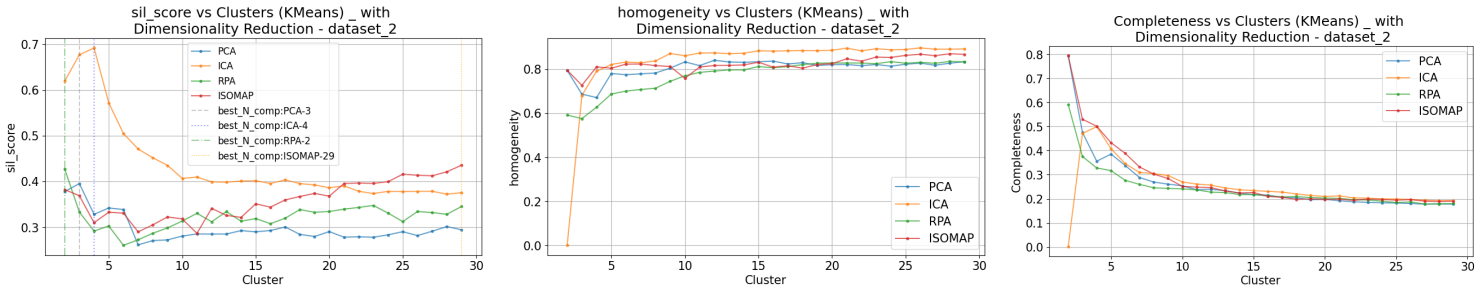
*1) **ICA**: fig (12a)*,Peak S' 0.117 was obtained at K = 14 using 12 ICA IC, which is lower than 0.23 of original data X, i.e the clusters are less separable and less cohesive. This is possibly due to loss of structure on transformation in IC variables, as ICA assumes independence and non-gaussian features, this could happen if these assumptions fail. Additionally ICA transformation may change the scale of features, which could result in different clusters with worse guaranties. H increases with K, therefore compared to the base KMeans model, each cluster has more similar Y labels,*fig (12b,12c,13a)*. At the same time, as explained previously, AMI and C decreases.

*2) **IsoMap**: fig (12a)*,Peak S' 0.6 was obtained at K = 2 using 12 ICA IC, which is significantly higher than the original data X, this may depict a manifold relationship between the features, not captured by the linear Transformations. However this could also be because of model overfit, as observed, these clusters have Least H and AMI, even though slightly higher C, these values are significantly low and close to 0,*fig (12b,12c,13a)*. ICA takes significantly higher time than IsoMap as the Ica projections are still very skewed and noisy.

B. Dataset 2

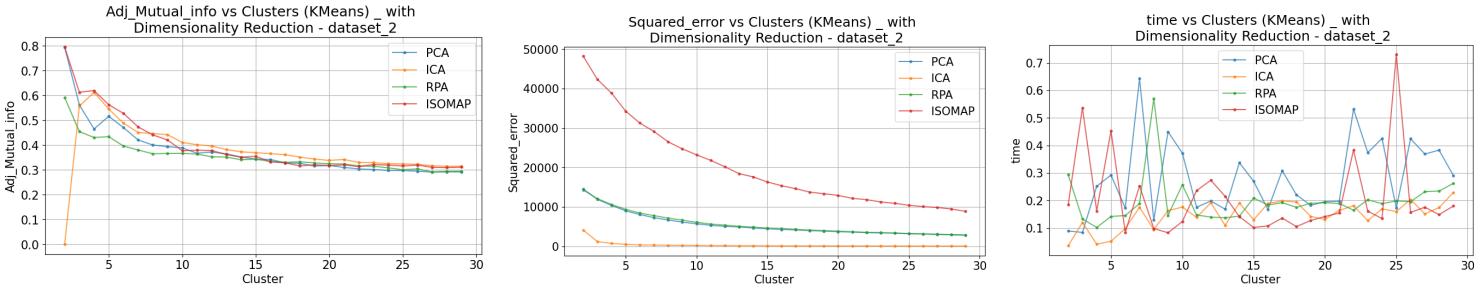
*1) **ICA**: fig (14a)*,,Peak S' 0.69 was obtained at K = 4 using 2 ICA IC, significantly higher than 0.35 of original data X, i.e Higher separability and cohesion in the as clusters. This is possibly due to reduction of noise in the cluster segregation using IC. This is further supported by higher H by 10% from base KMeans, that is instances of clusters have more similar labels. A slight increase in C and AMI, represent labels have more dedicated clusters and each clusters are less similar to each other,*fig(14b,14c,15a)*. Therefore ICA is particularly useful in the tasks of segregation where Target has associated structure with the data, possibly driving independence in IC. Further increase in K results in Low S',C and AMI, breaking the structure of the data.

*2) **IsoMap**: fig (14a)*,Peak S' 0.38 was obtained at K = 29 using Isomap obtained in previous section, higher than 0.35 of original data X, however not comparable to ICA, S' follows from high to low and then linearly increments thereafter. This is because IsoMap creates non-linear segregation to identify the similar instances, which follows a neighbourhood property resulting in similar projection of closer neighbours in this non-linear space, therefore with increase in K, more similar instances will be grouped. This also increase H, at the same time decreases C and AMI,*fig(14b,14c,15a)*. On average both ICA and IsoMap take similar time to fit.



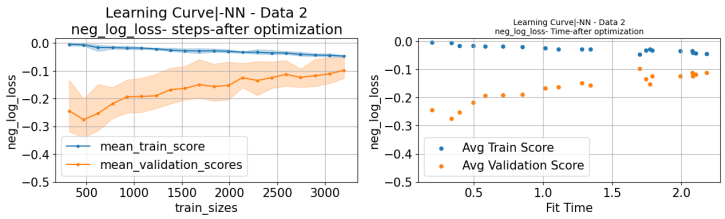
(a) Silhoutte Score Vs K (b) Homogeneity Vs K (c) Completeness Vs K

Fig. 14: K Means - Dataset 2 : Performance Measurement with Dimensionality Reduction

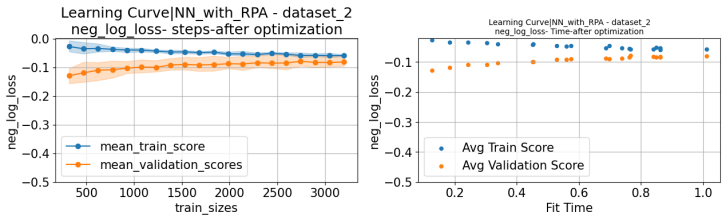


(a) Adj Mutual info Vs K (b) Squared error Vs K (c) Fit Time Vs K

Fig. 15: K Means - Dataset 2 : Performance Measurement with Dimensionality Reduction

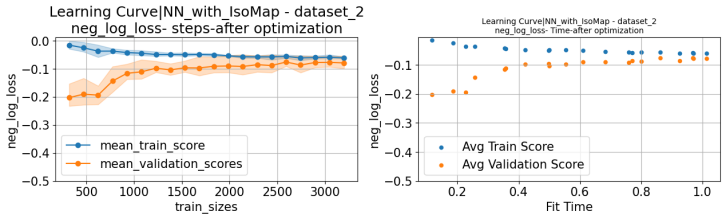


(a) Base NN with Scaled X: a.1 Train size, a.2 Fit time

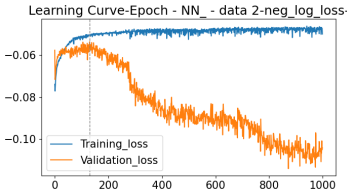


(b) With RPA projections: b.1 Train size, b.2 Fit time

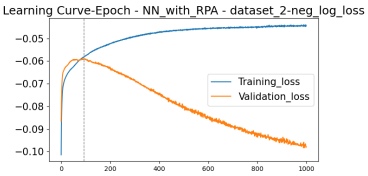
Fig. 16: Neural Network Learning Curve - Dataset2 : Performance Comparisons after optimization



(a) With IsoMap projections : a.1 Train size, a.2 Fit time



(b) Base NN



(c) With RPA projections

Fig. 17: Neural Network Epoch Learning Curve - Dataset2 : Performance Comparisons

VII. 4. Modelling Neural Network using Reduced Data - Dataset 2: NNRD

For this Analysis, a NN was tuned using gridsearch for each set of Reduced dimension data over Initial learning rate, Hidden layer size, iteration count and batch size. The choices were limited due to computation limitations, constrained to ensuring study of relevance of parameters. While Implementation was done over all Dim. Red. Algorithms, this section particularly covers RPA because of its interesting implications on performance without intricate work and IsoMap, Over Dataset 2, which provide better convergence guarantees due to its nature, to study different aspects of the model. The evaluation of the models is done using Negative log loss in addition of measuring Classification Homogeneity, completeness and AMI. Number of classes = Clusters these metrics are equivalent.

A. *RPA*

fig(16b,20),post tuning, NN procured Cross validation (CV) Negative Log Loss (NLL) - 0.081 and Train NLL -0.058, where prior is lower than original NN trained on scaled features, with CV NLL -0.097 and Train NLL -0.046559,*fig(16a)*, signifying lower overfit with the new features. The model converged faster in terms of iterations (92 epochs),*fig(17c)*, than original (130 epochs),*fig(17b)*, and 40% faster in avg. fit time. Additionally NN-RPA reaches a very low NLL very fast and shows comparatively smoother transition with Training size. As the resultant features are non-gaussian projections, the as possible error/ variance to overfit is less, which would result in such a fast performance. However this could also result in loss of certain information in transformation, creating a slightly more general model, which could be seen in test NLL of -0.059 compared to -0.053 in base NN. Therefore it provides a trade-off of stability, speed and performance.

B. *IsoMap*

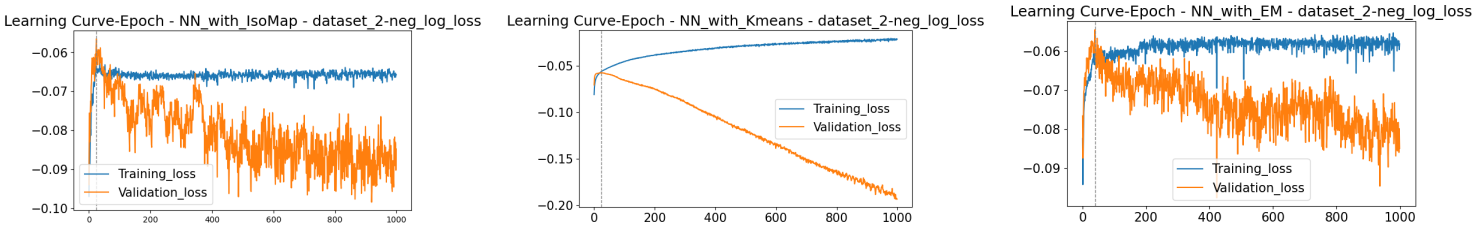
Capturing the manifold non-linear transformation, attained a CV NLL -0.079 and Train NLL -0.061,*fig(17a,20)*, comparable to RPA, however the close CV and Train NLL results in a more robust and balanced model, trading bias and variance. Additionally, the peak performance convergence was achieved much faster at 24 epochs*fig(18a)*, i.e 20% of base NN, with average fit time 40% less than the later even with lower batch size, similar learning rate and Hidden layer size, IsoMap projection transforms features to metric of closeness of non-linear spaces, therefore features inherently are similar to similar instances, which also results in less number of weight updates to identify the boundary of such segregation resulting in faster training. Test NLL of 0.056 was observed, though lower than but is closer to base. Another effect of such a transformation can be seen from Learning curve over iterations, where this transformation resulting in errors decreases the effect of overfitting. Additionally H,C and AMI 0.838*fig. 20)*, i.e predicted class over training data is very close to the true class. Therefore, when the data has inherent quality of segregation, the domain knowledge can be used to create features resulting in more robust and faster models.

VIII. 4. Modelling Neural Network incorporating Data Clusters - Dataset 2: NNC

Similar to the previous section, NN was tuned to convergence over dataset2, however using Clusters identified in Section IV as new features in addition to original features. Since the number of Clusters identified by both the algorithms are less, it is convenient to represent them as one-hot encoding due to categorical nature. For 3 clusters, 2 one hot columns can be created without loss of information. These features were concatenated with X.

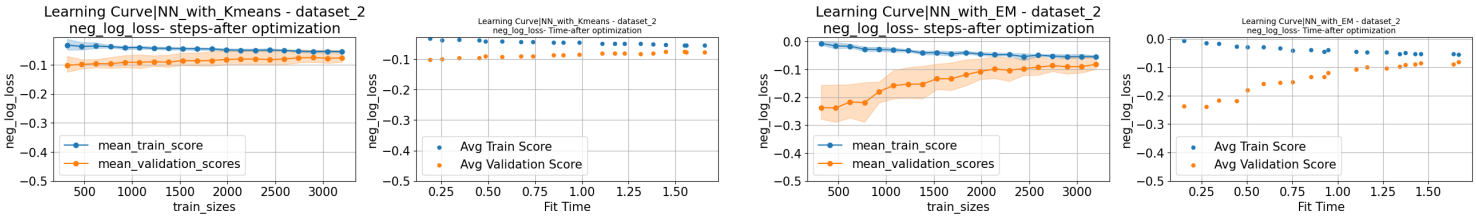
A. *KMeans*

3 clusters resulting in 2 columns were added to 7 original features. With CV NLL -0.076 and Train NLL -0.054,*fig(19a,20)*, the model performed better at CV than all the other models and while reducing model overfit. This is further supported by High H, C and AMI 0.842, predicted labels are have low l1 and l2 error and Not similar to each other. This additional information of model segregation results in less number total updates in weights to identify the decision boundaries, similar to IsoMap in 24 epochs,*fig(18b,20)*, even with smaller learning rate , however due to larger Hidden Layer size, every fit takes more time due to more weight updates, which would also depict the ability to identify more intricate patterns created due by addition of new cluster information. Due to this architecture the impact of the model overfit is more prominent over iterations.



(a) With IsoMap Projections (b) With Kmeans Cluster projection (c) With EM projection

Fig. 18: Neural Network Epoch Learning Curve - Dataset2 : Performance Comparisons



(a) With Kmeans Projections: a.1 Train size, a.2 Fit time (b) With EM Projections : b.1 Train size, b.2 Fit time

Fig. 19: Neural Network Learning Curve - Dataset2 : Performance Comparisons after optimization

B. *EM*

with only one feature addition representing 2 clusters creates the model closes to the Base NN, Smalled Train NLL of -0.0545 *,fig(19b,20)*, smallest among modified features NNs, CV NLL - 0.082, worst among modified feature set, and smallest Test NLL of -0.54425 after Base NN. Although with only 1 feature, the resultant model is less overfit, due to additional ability to generalize the classes based on cluster information. This is because, the additional information results a simpler hypothesis space of single layer HD (32,) and a batch size of 8. And so it also results in faster convergence in number of iterations, as the hypothesis space is smaller, in 39 epochs,*fig(18c,20)*. Therefore extracted segregation of data based on gaussian clusters also results in performance boost in term of robustness and speed boost.

NN Model performance comparison					
	Kmeans	**EM**	**RPA**	**ISOMAP**	**NN_Scaled**
mean_train_score	-0.054913	-0.054499	-0.058523	-0.061078	**-0.046559**
mean_validation_scores	**-0.076394**	-0.081853	-0.081195	-0.079365	-0.097131
Test_scores	-0.057601	-0.054425	-0.059048	-0.056533	**-0.053641**
mean_fit_times	1.655757	1.668196	**1.01216**	1.014304	1.698176
homogeneity	**0.84251216**	0.82803531	0.81750412	0.8387825	
Completeness	**0.84260633**	0.82815204	0.81751915	0.83890075	
Adj_Mutual_info	**0.84253084**	0.82806266	0.81747871	0.83881255	
batch_size	8	8	32	8	16
hidden_layer_sizes	(64, 32),	**(32,),**	(64,),	(64,),	(64,),
learning_rate_init	0.0005623	0.0177828	0.0031623	0.0177828	0.0100000
Max Iterations	**24**	39	92	24	130
Feature count	9	8	7	7	7

Fig. 20: NN performance comparison and Structure comparison

IX. Hypothesis Validation

1 Dataset 2 Due to its inherent segregation would result in low (close to two) best clusters.
- using only KM and EM, produced (3,2) clusters however with DR, (3,4,2,29).

2 Dataset 1 Due to its skewed nature will be difficult to cluster as it is, and therefore may required higher number of cluster to better represent the data.
- Low H,C and AMI throughout.

3 Dataset 2 Due to its inherent segregation and more binary variables would result in low number of projections to represent the data.
- Not enough reduction (5,2,7,7) 4 DRs

4 Dataset 1 due to more continuous variables and complex feature distributions would not result in significantly low projections to represent the data.
- Not enough reduction (7,12,11,8) 4 DRs

5 Dataset 2, Clustering with DR will not add much performance compared with raw scaled data
- Higher in H C and AMI with DR

6 Dataset 1 would be better clustered with projections than raw scaled data.
- Lower H C and AMI with DR

7 Dataset 2, NN with DR with possibly reduced number of features would converge faster and with less overfit
- Lower fit time and less difference in CV and Train NLL

8 Dataset 2, NN with Clustering will result in more sophisticated model, to capture new segregation in the data, and result in slightly longer training time however better performance.
- Kmean HD layer size increase, EM HD Layer size decrease, similar Fit time to base, lower performance.

X. Conclusion

This Study explores the Working of different clustering techniques and Dimensionality Reduction methods on two different type of data, highlighting key driving factors of each along with important levers to tune them. A major insight showcase the importance of dataset description in successful utilization of such techniques, along with the predefined goal, which governs the selection of algorithms, evaluation metrics and tuning. Additionally the study highlights the alignment of distribution and structure of noise in the data to make use of preference bias of techniques which use gaussian distribution.

References

[1] Wine quality dataset, Kaggle, www.kaggle.com/datasets/rajyellow46/wine-quality
[2] Gender Classification dataset, Kaggle, www.kaggle.com/datasets/elakiricoder/gender-classification-dataset.