Assignment#3, CS7641

Georgia Institute of Technology, Fall 2020

Asif Rehan, arehan7@gatech.edu

1. Introduction

Clustering helps identify groups in a dataset in an unsupervised learning. Also, being able to reduce the dimensions of the data may help eliminate noise, reduce data size, improve runtime, and increase performance.

In this assignment, KMeans and Expectation Maximization clustering algorithms are applied on two datasets. Also, three Dimensionality Reduction (DR) algorithms, Principal Component Analysis (PCA), Independent Component Analysis (ICA), Random Projections (RP), and a feature selection technique using Random Forest Classifier (RFC).

2. Data

From assignment#1, **Wine Quality Dataset** involves predicting the quality of white wines on a scale given chemical measures of differentwinesamples and labels them by quality rating of 0-10. It is a multiclass problem to predict the quality rating of the wine based on the chemical characteristics. The dataset is not balanced, and it comes with 4898 data points. There are 11 input variables.

Second, The dataset comes with different input variable with medical condition indicators such as age, BMI, insulin etc and the Outcome column, which is 1 if the target had diabetes or 0 otherwise, making it a binary classification problem. There are 768 records with 8 variables and 1 label column. This data is a bit imbalanced, there are 500 false (~65%) and 268 (~35%) true diabetes indicators. Visually no outlier was detected, and all features were numerical, so no feature engineering was required. The scatterplot shows there are a few outliers handled by preprocess step with standardization.

3. Methodology

First, the two clustering algorithms were applied on the two datasets, Wine Quality and Pima Databetes dataset.

Second, four dimensionality reduction algorithms were applied to the two datasets.

Third, clustering experiments in the first step were repeated on 16 combinations of datasets, dimensionality reduction, and clustering method, but on the data after running dimensionality reduction on it.

Fourth, Neural Network classifier was run on the newly projected data from step 4 above for Pima data. So four Neural Network learners were created for each of the four dimensionality reduction algorithms.

Fifth. Neural Network classifier was run on the enhanced Pima dataset by adding above for Pima data. So two Neural Network learners were created for each of the clustering algorithms.

3.1. Code used

For the implementation of the algorithms, Python library Scikit-Learn is used.

3.2. Train-Test-Split

The data was scaled using min-max scaler based on the cluster data set to scale between 0-1 for each feature before using. Then data is split into three parts, cluster (35%), Neural Network (35%) and hold-out testing data (30%). Out of all the data, 35% was used to build the clustering and Dim. Red. algorithms, 35% was of the data was used for training Neural Network using the cluster and DR algorithms. This is to reduce data pollution of neural network training set with the cluster data. The hold out testing set is used only

4. Analysis and Findings

4.1. Initial Clustering

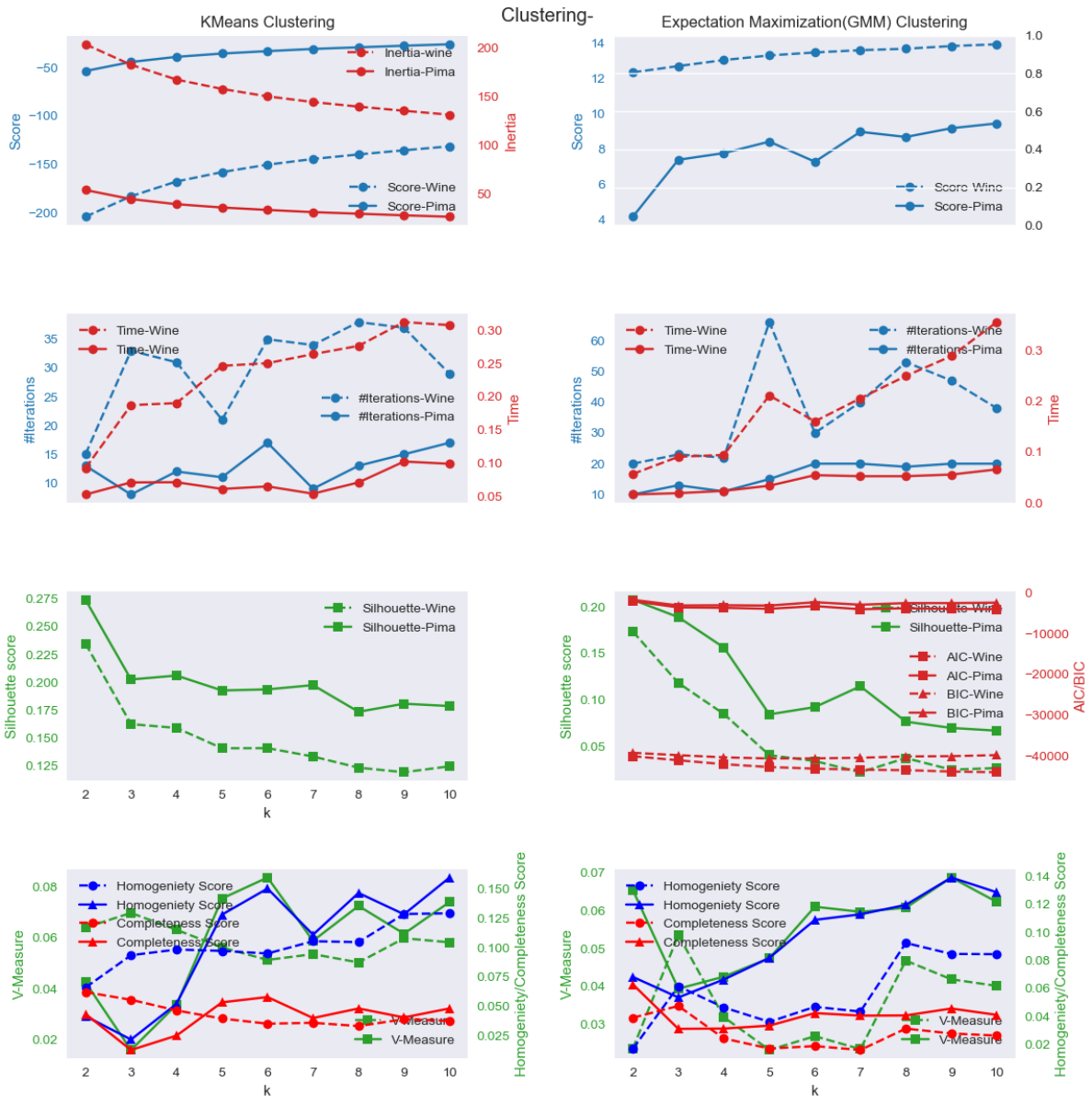
For, K-Means a range of values from 2 upto the number of features for each dataset was used as the k-value. The pairwise distance function is Euclidean. The 'k-means++' was used which selects initial cluster centers for k-mean clustering to speed up convergence. Maximum 300 iterations were allowed. For each k-value,

For Expectation Maximization (EM), *GaussianMixture* model from Scikit-Learn was used. a range of values from 2 upto the number of features for each dataset was used as the k-value.

The chart below show the metrics for a range of k values. On the left column of chart, metric of Kmeans for both wine and pima data are shown. On the right, for EM.

For the Kmeans's metric in the left column, in top left, Score for Kmeans is within-cluster sum of squares (WSS). For both pima and wine the best and lowest WSS value is for 2 clusters (k=2). Also the Inertia values also are highest for k=2 for both since they are binary class data.

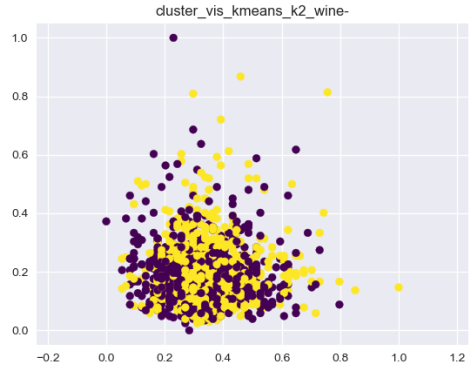
Second from top, left, the runtime and iterations are not linearly related to k. Third from top, left, the silhouette scores are also the best for k=2 for both pima and wine data. A value close to 1 implies that the instance is close to its cluster is a part of the right cluster. Whereas, "a value close to -1 means that the value is assigned to the wrong cluster" [ref]. Clearly with more clusters, the data may not be clearly separable. Bottom top, the homogeneity score and completeness score are not very great and so is the V-measure which is the combined score.



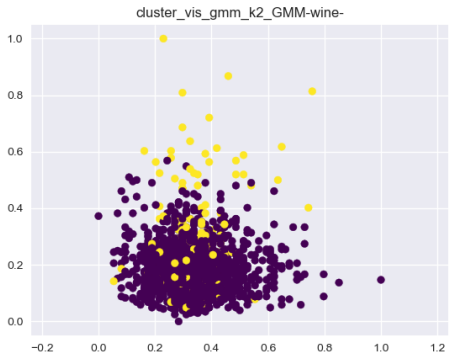
On the right column, the metrics for EM is shown. Top right, the score for EM is Log likelihood of the Gaussian mixture. The likelihood is the most with higher clusters. But the silhouette score, in right, third from top, shows is best for k=2 for both wine and pima. EM does better with arbitrary gaussian surface when Kmeans's spherical boundary does not do well. But here is seems, both does pretty well for k=2. In bottom right, the homogeneity score and completeness score are similar to that of Kmeans', so does the V-measure.

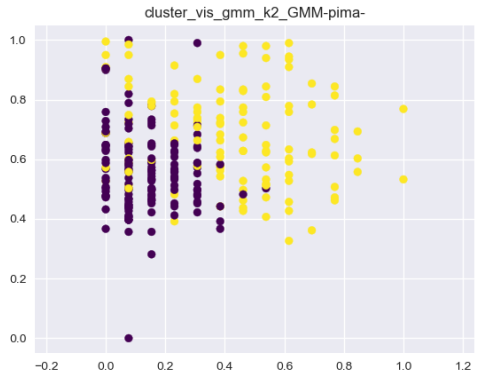
The two charts below show the basic clustering results for the K-Means clustering algorithms. Using the first two columns in the dataset for Wine and Pima, the charts show the clusters in Wine are not as clear as in Pima dataset. Similar observation was already made in Assignment#1, that the Wine quality data is originally a multiclass problem which has been relabeled for binary classification based on the quality

value threshold. So the data is not clearly binary classifiable. However, Pima data does originally refer to a binary classification problem. So the data performs better.



Similar observation were made for GMM clustering below. The clusters are more overlapping for GMM and the linearity of the decision boundary for Pima data is more prominent compared to KMeans. This shows the capability of GMM to take a non-circular region.



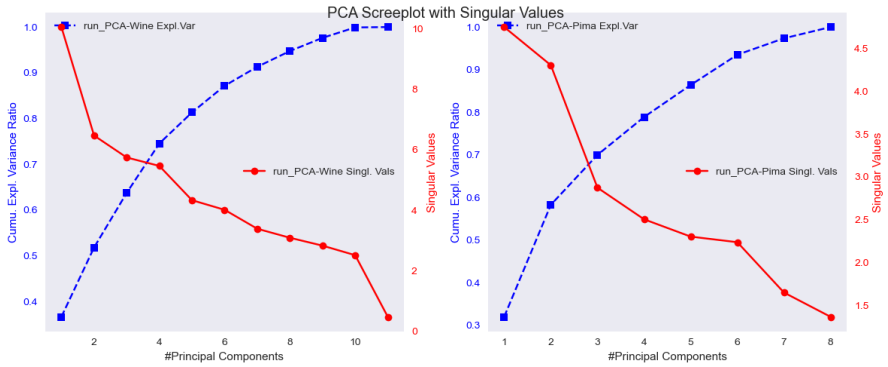


The data seem to make sense as the chart shows. However, the Homogeniety and compleness measures penalized the predicted labels not aligned well with the original binary data. So the scores were low. Further data cleaning such as outlier elimination could help, though 0-1 scaling should already take care of it mostly.

4.2. Dimensionality Reduction and Feature Selection

4.2.1. PCA

The SkLearn's PCA module was used. The PCA was done by using Singular Vector Decomposition. For preprocessing, whitening of the data was applied which basically applies normalization of the data which helps with creating Gaussian spherical representation of the data.



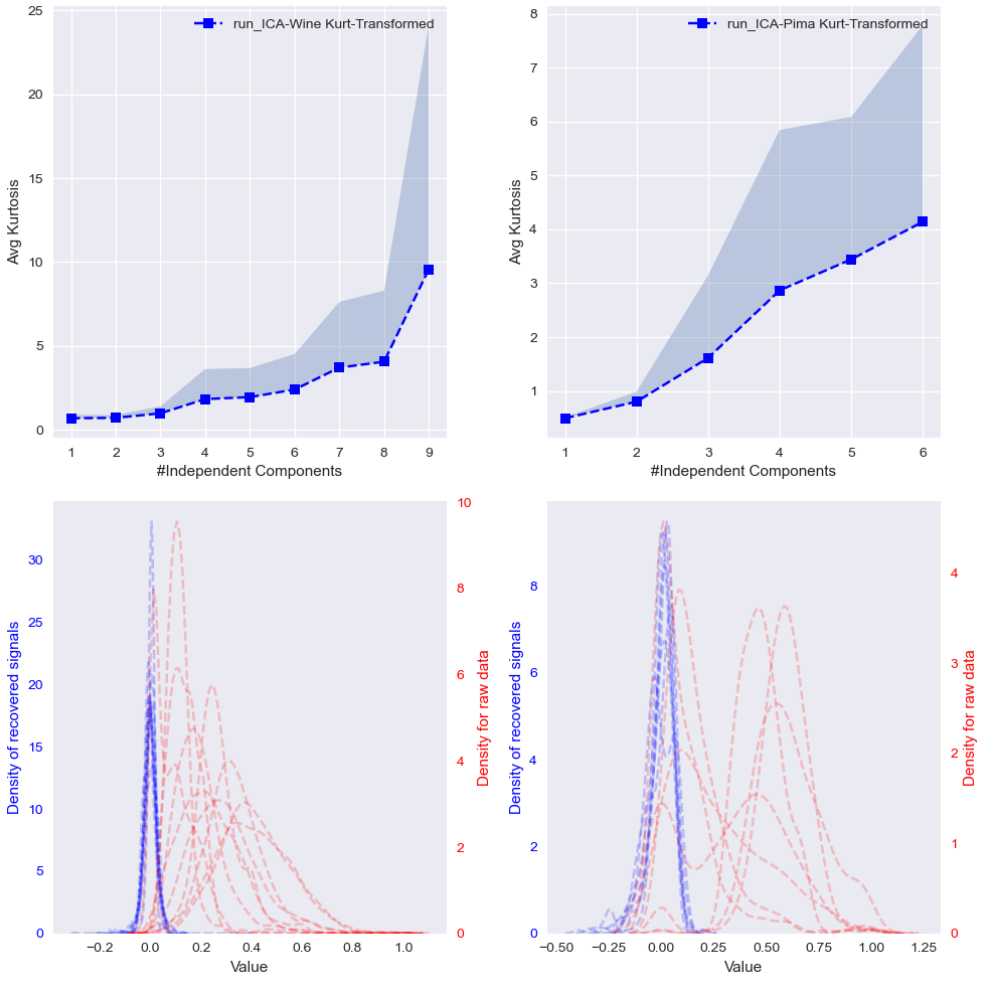
The scree plots below shows the results for Wine data on left, and Pima data on the right. The blue line shows the Cumulative Explained Variance Ratio on the blue Y-axis. The Singular values are show in red line following the red ticks on the right side of the chart.

As more and more principal components are added, Explained Variance Ratio reaches almost 1, which means the original data is retained in its entirety but just transformed. Also the principal components are sorted by importance by PCA, as a nice feature.

To select the best number of principal components, those first components were used that has cumulative 95% explained variance ratio.

1.1.1. ICA

FastICA module from SkLearn was used. Again, for preprocessing, whitening of the data was applied which basically applies normalization of the data which helps with creating Gaussian spherical representation of the data.



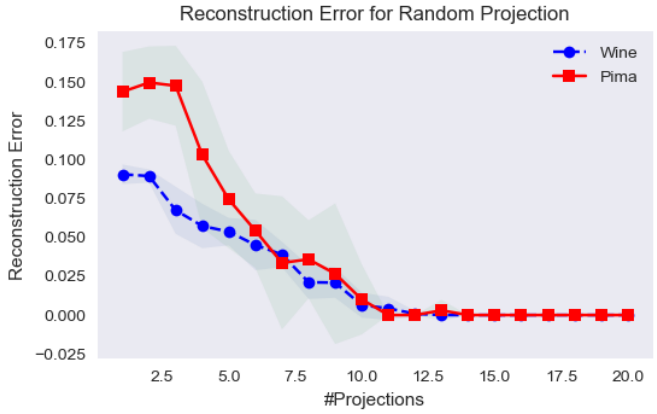
A range of number of possible Independent Components (IC) were used to build different reduction. However, to select the best number of Independent Components, the one which appears to fit the best is selected by looking at average Kurtosis value. From this chart below, the top chart on the left show average kurtosis for each IC number for Wine dataset, and on the right shows for Pima dataset. The shaded regions show the mean+std dev of the values. Apparently more Independent components produce better kurtosis. Since with ICA the observed signal data are gaussian and the recovered IC's are supposed to be independent and non-gaussian, so their kurtosis is supposed to be higher than 0 for Fisher's kurtosis definition which defines Gaussian distribution to have 0 kurtosis.

Also the bottom row of charts show the Kernel Density Estimation plots for Wine data (left) and Pima data (right). The IC's KDE are scaled on the left axis in blue and the original features' KDE values are shown in right axis. As the top chart shows, the recovered IC's have higher kurtosis (kurt~=30 for wine, 10 for Pima data) than the original signals (kurt~=3).

For wine, selecting 9 IC's would not reduce the data from original 11 features. So 8 IC's are selected. For Pima, 6 IC's was selected to reduce from original 8 features.

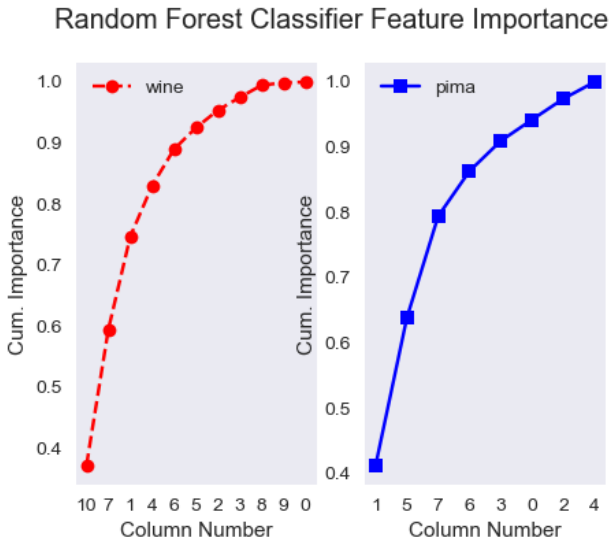
1.1.1. RP

To see variability in the Random Projection algorithm for DR, the projections were repeated for a set of seeds. The SparseRandomProjection module from sklearn was used. A range for the projection was tried and the reconstruction error were averaged across multiple runs. The results show the reconstruction error drops as more random projections are added. Interestingly, for Pima the reconstruction error increases up to the first three projections and then drops. This indicates the projections are not ordered by importance. Form this chart, 8 projections for Wine data and 6 for Pima data was finally used. 8 because, wine data had 11 features already and Pima data did have 8 features.



1.1.1. RFC

Using Random Forest Classifier as a feature selection tool, feature number 10,7,1,4,6,2,5 were used for Wine data and 1, 5, 7, 6, 3, 0 were found to be most importance based on the Feature Importance values from the RandomForestClassifier module in sklearn.



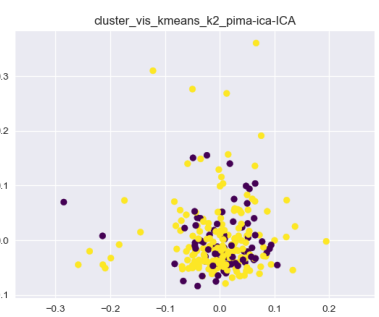
Only the features with Cum. Importance less than or equal to 0.95 were kept to reduce the data. 100 weak learners were used and 10% of the sample size is set as the leaf size for each data. From the models, the

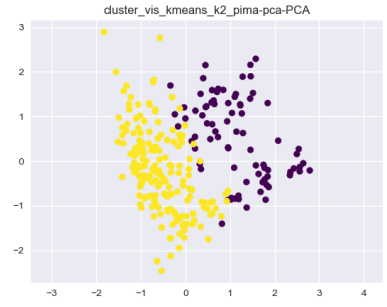
features are ranked by Cumulative Importance in the chart below.

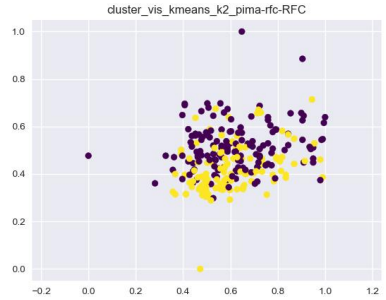
4.3. Clustering the Reduced Datasets

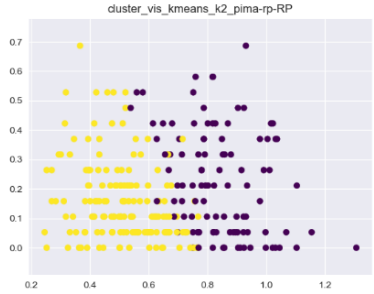
At this step, the reduced data sets were fed into clustering. Total 16 combinations of the datasets, DR, clustering algorithm were used. Out of that, the below charts were the most interesting. They are for Pima Data with k=2, for all 4 DR algorithms for Kmeans clustering only. The color shows the two classes predicted by the clusters.

From the charts we can see, the ICA and RFC did not do that well. PCA and RP were better contenders. This is a confirmation of the previous observation above even before the data was Dimensionally Reduced. However, the charts only show two columns of the data. A high dimensional image could help comprehend the phenomena better.

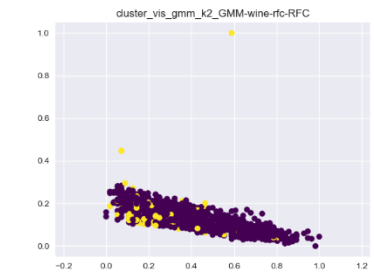


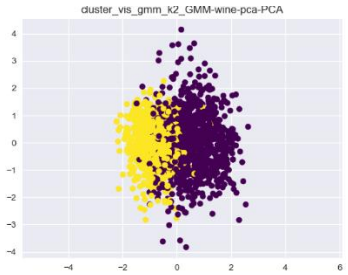


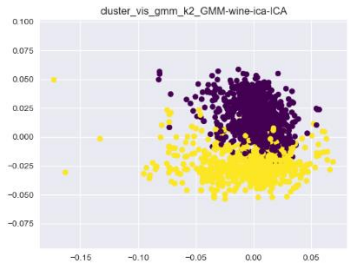


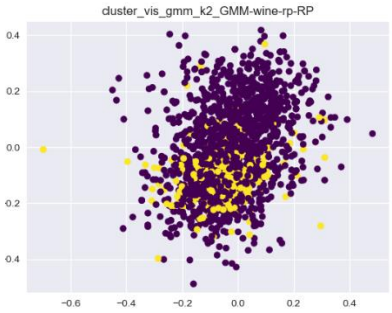


More interestingly, the Wine data shows interesting pattern for GMM models using the DR methods. The first two dimensions in RFC actually make is an elongated ellipse. While PCA and ICA makes the classes two circular regions.









However, these transformations did not hep the V-score measures that much because the clustering process could not apparently learn the true complex boundaries.

4.4. Neural Network Classification of the Reduced Datasets

For the 4 DR algorithms, the DR model is used on the Pima dataset.

For the neural network classifier, a network with (12, 6,4,2) hidden nodes, with adaptive learning rate, and a logistic activation was used. If a network is stuck at local optima, maximum 100 iterations are watched allowed with 80 iterations for early stop with no improvements.

For grid search over all 4 DR algorithms, 2 clustering algorithms and a benchmark GridSearch was employed to find the best activation between logistic and tanh, best L2 regularization.

For the benchmark NN, the testing data for NN was used, only scaled by MinMax between 0-1.

For the DR algorithms learned from the cluster train data, the learned DR model is used to reduce both NN-train and holdout test data. These data were used to train the NN for DR datasets.

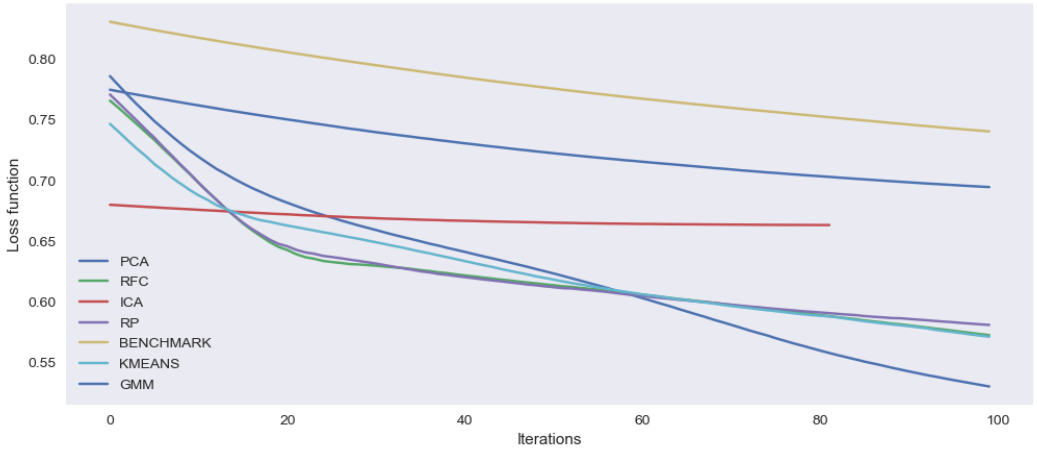
The results are in next section.

4.5. Neural Network Classification of the Enhanced Datasets using Clusters as New Features

The Pima dataset was also used for this step.

For the two clustering models learned from the cluster train data, the NN-training data was scaled and then the labels are predicted using the models. Then the labels are added to the original NN-train data as an extra feature. The hyperparameter tuned models are then used for training.

The 4 DR, 1 benchmark and 2 clustering approaches for supervised NN classification are show in the table below. From the 7 data, it shows the PCA models has the sharpest Loss reduction, whereas ICA approach did not learn much and did have a flat loss curve. RP and RFC were very similar. So the PCA approach was the clear winner as it could learn the fastest among the others.



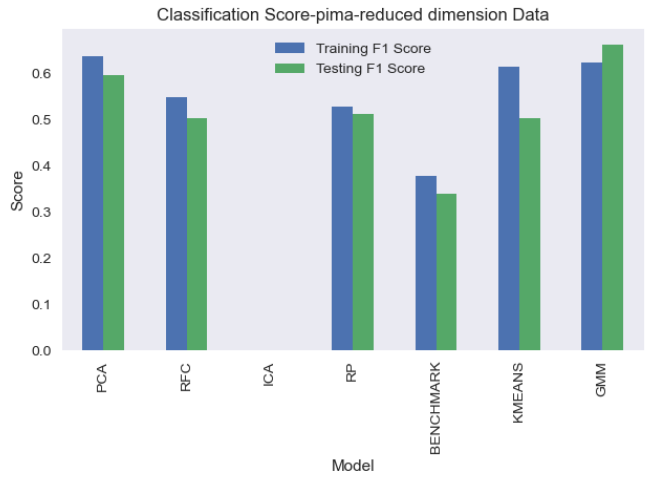
In the table below, the results show the GMM-cluster based enhanced data had the best F1 score for testing. But PCA-based DR approach resulted in the best training F1 and the second best testing F1 score.

The dimensions of the Pima training data for each approach is showed in the Dimension column, notice the benchmark Pima data had 8 features. The Kmeans and GMM approach enhanced the dimension by 1

while DR models reduced the dimension. This reduction in dimension did not reflect on model performance directly except for GMM and PCA from the clustering and DR groups respectively.

	Model	Dimension	Training F1 Score	Testing F1 Score	Training Time(sec)
0	PCA	7	0.635294	0.595238	0.146608
1	RFC	6	0.548780	0.503226	0.146641
2	ICA	6	0.000000	0.000000	0.383489
3	RP	6	0.528302	0.510638	0.220410
4	BENCHMARK	8	0.377432	0.338583	0.201461
5	KMEANS	9	0.614525	0.502994	0.229387
6	GMM	9	0.622568	0.661417	0.183510

The chart below shows the F1 scores more clearly.



Clearly ICA model did not learn at all as seen in the Loss curve above. So its F1 for train and test was 0. A different hyperparameter might be able to help or a different number of IC's might help it. The backpropagation might have been stuck a local optima from the initial stage and could not recover.

As the data is reduced the run time change compared to benchmark did not show clear relationships. However the data size reduction could be important which could not be considered for this assignments.

5. Conclusions

From the analysis, DR and clustering are powerful tools to enhance supervised learning by using unsupervised learning methods. The PCA is very powerful and K-Means and EM based clustering are useful for regular unsupervised data labeling.