CS 7641 – Unsupervised Learning

mgupta318

Abstract

The paper has been structured into five sections where the first section describes the clustering algorithms, second section describes about the dimensionality reduction algorithms, third section describes running the clustering algorithms again using the dimensionality reduction data, fourth section describes about running the neural network on the second dataset using the projected data from dimension reduction algorithms and fifth section describes about adding the cluster labels as new features to the dimension reduction projected data and running neural network. Training times, metrics for comparison scores between original labels and new dimension reduction labels and accuracy scores for neural network algorithm with all dimension reduction projected data are published through various experiments. The clustering algorithms used are K-Means and Expectation Maximization (EM). The dimensionality reduction algorithms used are Principal Component Analysis (PCA), Independent Component Analysis (ICA), Randomized Projection (RP) and Linear Discriminant Analysis (LDA). sklearn library has mostly utilized for running the algorithms used for this paper.

Datasets

1. White wine quality–This dataset has multiple features, which help in predicting the wine quality. This is an interesting classification problem where the last column 'quality' contains the values from 0-10. This column was transformed to a classification problem into 0 and 1 values where wine quality less than 6 was encoded with label 0 and others with label 1. This dataset is an imbalanced dataset since the proportion of y labels was in the ratio of 33:67 (0, 1 labels) with 4898 samples, 11 features and 1 target label. After scaling the number of features remain the same since all were numerical. A split of 80:20 was used between train and test dataset.
2. Adult Income Dataset–This dataset has multiple columns related to an individual like age, state. The target column is income, which is classified as a binary classification problem. This dataset had both categorical and integer data which was transformed using encoding into numerical data. This is useful in predicting the income of an individual based on its different attributes. This dataset is an imbalanced dataset since the proportion of y labels was in the ratio of 76:24 (0, 1 labels) with 6543 samples, 14 features and 1 target label. After scaling the number of features increased to 106 since it was scaled from categorical data to numerical data. sklearn train test split was used for splitting into train and test data.

Both the datasets are classification problems and have been reused from Assignment1.

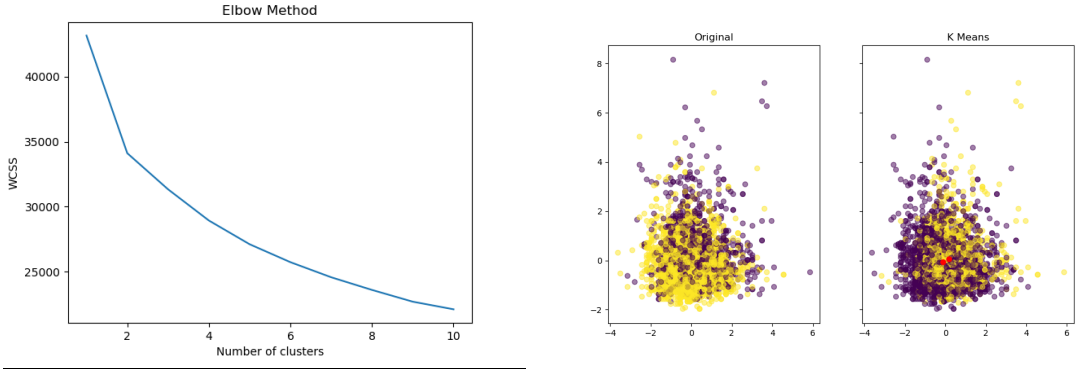
Section1 – Running the clustering algorithms

The two clustering algorithms used for this paper are Kmeans and Expectation maximization. These algorithms were run on both datasets.
Kmeans clusters the labels based on the average mean of the similarity points and recalculates centroid every iteration till the point the points classification is not changing for the cluster assignment. The optimal value of k is calculated by using WCSS score providing a range of clusters.
Expectation maximization calculates the probability using the Gaussian mixture and accordingly assigns clusters to the data points. The optimal value of components is calculated by using Silhouette score providing a range of clusters.

The value of k (number of clusters) from the Kmeans was computed from the Within Cluster Sum of Squares (WCSS) score, which provides the squared average distance of all points within cluster to centroid using Euclidean distance. Expectation Maximization uses Silhouette score, which is derived from the mean, inter cluster distance and mean nearest cluster distance for each sample.

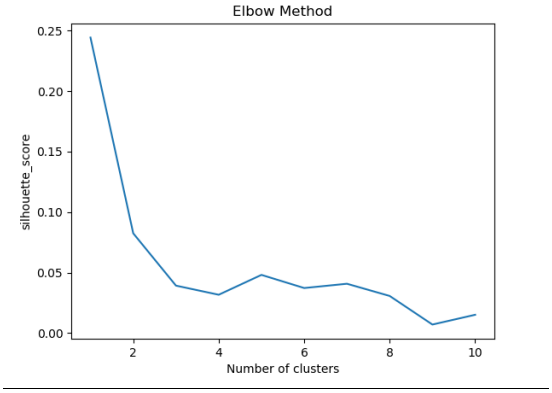
Dataset1

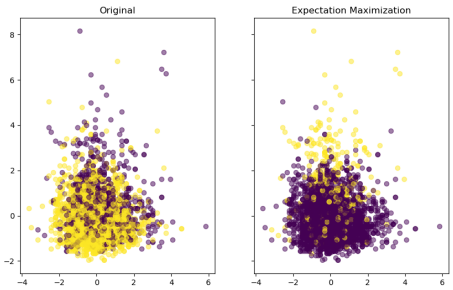
a) Kmeans



As seen from the WCSS graph the number of clusters identified were two. Kmeans algorithm was run using two clusters and it places two centroids (red dots) clustering the data.

b) Expectation Maximization

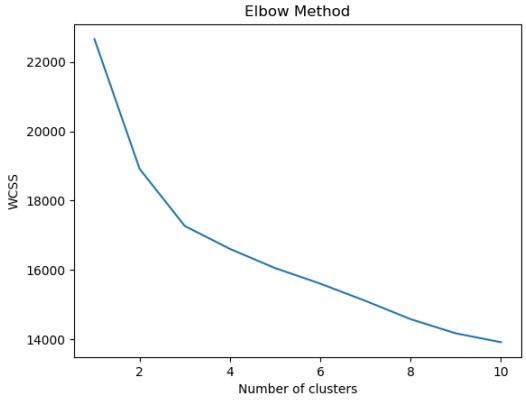


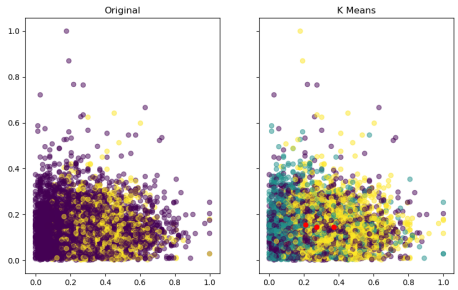


As seen from the Silhoutte graph the number of clusters identified were two. EM algorithm was run using two clusters and performs clustering the data.

Dataset2

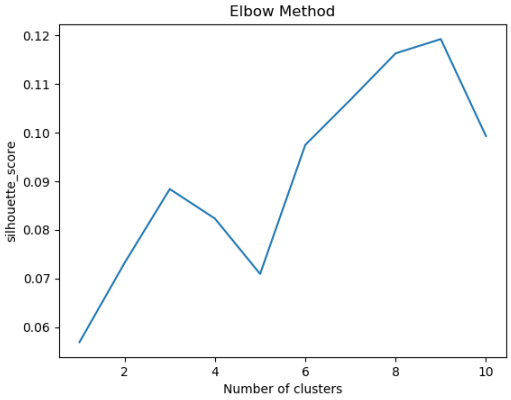
a) Kmeans

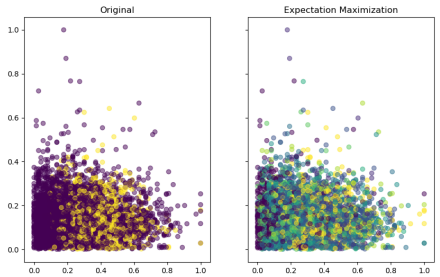




As seen from the WCSS graph the number of clusters identified were three. K-means algorithm was run using three clusters and it places two centroids (red dots) clustering the data.

b) Expectation Maximization





As seen from the Silhouette graph the number of clusters identified were nine. EM algorithm was run using nine clusters and performs clustering the data.

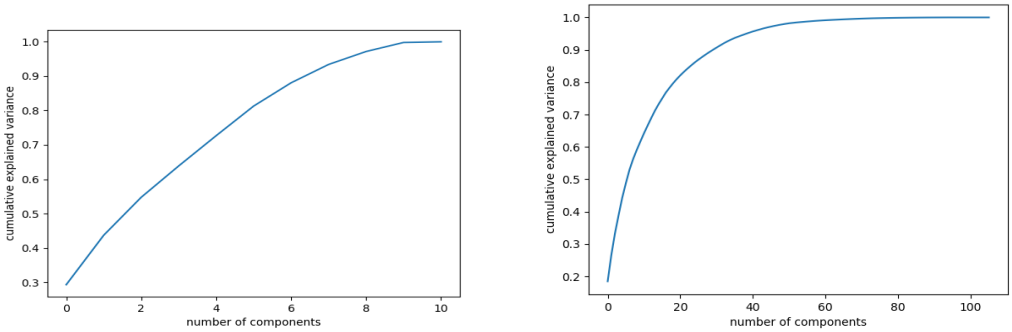
The data is not clustered in a good way when we run the original Kmeans or EM algorithms on both these datasets and we will apply the dimensionality reduction for individual datasets in next section to see if the clustering improves. The reason of this can be both these datasets are classification problems and not clustering problems.

Section2 – Running the dimensionality reduction algorithms

The below four dimension reduction algorithms were run on both the datasets:

1. Principal Component Analysis (PCA) – PCA works by transforming the features of the input into principal components with a goal of maximizing the co-variance between different features. The principal components selected are orthogonal in nature. Depending on the eigen values for each PC the ones with minimum variance are dropped and ones with maximum are preserved. The shape of the explained variance ratio provides the number of transformed dimensions. Explained variance ratio provides the number of transformed components for the eigen values of the principal components. PCA internally does transformation and then feature selection also.

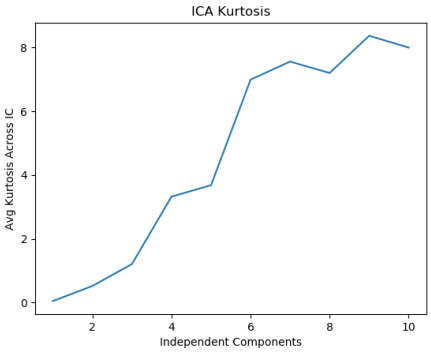
Dataset1 **Dataset2**

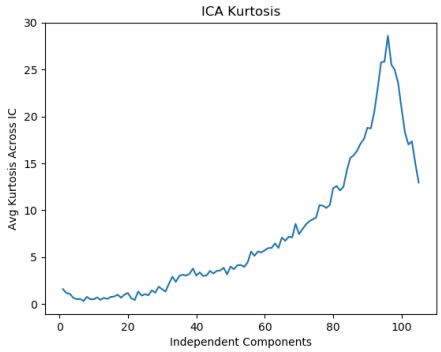


For dataset1 the number of components selected were 9 and for dataset2 it was 50. This was selected based on the fact that using these components preserved more than 95% of the co-variance.

2. Independent Component Analysis (ICA) – ICA works by transforming data based on the mutual independence between the features. Kurtosis score was calculated which shows the number of components to be selected in transformation based on the mutual independence score.

Dataset1 **Dataset2**





For dataset1 the number of components selected were 9 and for dataset2 it was 95. The average kurtosis score initially increases and then starts decreasing. The point where the average kurtosis score starts decreasing was selected as the number of components.

3. Randomized Projection (RP) – Randomized projection works by selecting the random projections and transforming data into linear space by using the pairwise distance between two points using Euclidean distance. The number of components was selected based on running RP several times and observing the variance of the original data with the projected data.

4. Linear Discriminant Analysis (LDA) – LDA assigns the Gaussian density to each target label and then transforms the dimensionality of the data to the most discriminative directions. The number of components selected in LDA was two and the data was reduced to one dimension.

Section3 – Running clustering algorithms on the new projected data

After getting the new projected data from four dimension reduction algorithms both Kmeans and EM were run on projected data each from PCA, ICA, RP and LDA. Before running the clustering algorithms again the number of clusters were recomputed using the WCSS score for Kmeans and Silhouette score for EM. The value of k obtained was used on re-running the clustered algorithms. The comparison scores like Homogeneity - if data points are members of single class, Completeness – class and cluster values are same for data point, V-measure, Rand

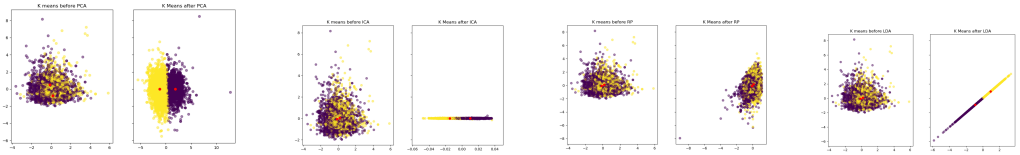
Score – computes similarity between two clustering and Mutual Info – computes mutual independence between two classes were computed for both the train and test labels after running clustering algorithms on the projected data and comparing against the original train and test labels. The

Dataset1

Kmeans:

Running Kmeans on the projected data

PCA **ICA** **RP** **LDA**



Comparison scores

Kmeans train					
Scores	**Original**	**After PCA**	**After ICA**	**After RP**	**After LDA**
Homogeneity	0.044	0.044	0.092	0.055	0.142
Completeness	0.042	0.041	0.087	0.052	0.131
V-measure	0.043	0.043	0.089	0.053	0.137
Rand Score	0.078	0.077	0.059	0.087	0.151
Mutual Info	0.043	0.042	0.089	0.053	0.136

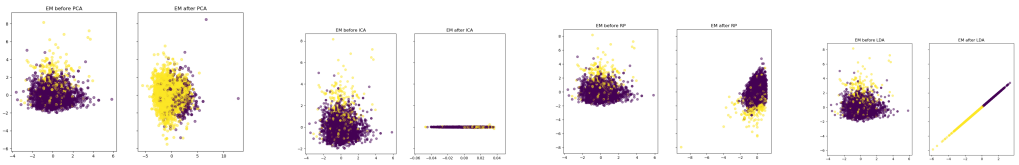
Kmeans test					
Scores	**Original**	**After PCA**	**After ICA**	**After RP**	**After LDA**
Homogeneity	0.03	0.03	0.096	0.052	0.148
Completeness	0.029	0.028	0.089	0.047	0.135
V-measure	0.029	0.029	0.092	0.05	0.141
Rand Score	0.06	0.058	0.063	0.068	0.15
Mutual Info	0.029	0.028	0.092	0.049	0.141

The number of clusters was chosen again for each dimension reduction algorithms by calculating the WCSS score. The value of the clusters obtained was same as original which are 2. PCA, ICA and LDA were able to cluster the data very well into two individual clusters as shown in the graphs above with centroids clearly clustering the data. RP did not perform very well for the clustering in this case. The above comparison scores states how various dimension reduction algorithms lined up with the original labels which shows LDA did the best but the labels were not that closely related with the clusters.

EM:

Running EM on the projected data

PCA **ICA** **RP** **LDA**



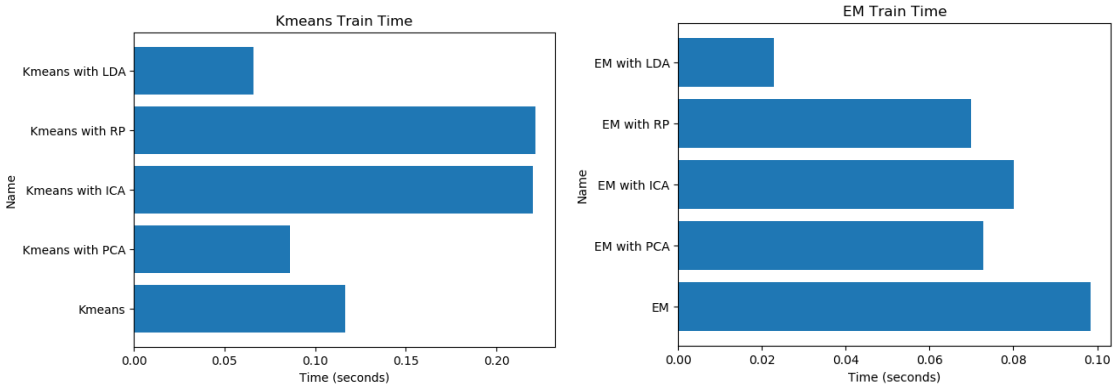
Comparison scores

EM train					
Scores	**Original**	**After PCA**	**After ICA**	**After RP**	**After LDA**
Homogeneity	0.032	0.048	0.048	0.024	0.139
Completeness	0.059	0.07	0.07	0.046	0.129
V-measure	0.042	0.057	0.057	0.032	0.134
Rand Score	0.083	0.115	0.115	0.07	0.127
Mutual Info	0.041	0.057	0.057	0.032	0.133

EM test					
Scores	**Original**	**After PCA**	**After ICA**	**After RP**	**After LDA**
Homogeneity	0.034	0.055	0.057	0.028	0.13
Completeness	0.068	0.083	0.087	0.054	0.119
V-measure	0.045	0.066	0.069	0.037	0.124
Rand Score	0.087	0.129	0.133	0.079	0.11
Mutual Info	0.044	0.065	0.068	0.036	0.123

The number of clusters was chosen again for each dimension reduction algorithms by calculating the Silhouette score. The value of the clusters obtained was same as original which are 2. The above comparison scores states how various dimension reduction algorithms lined up with the original labels which shows LDA did the best but the labels were not that closely related with the clusters.

Training time comparison between Kmeans and EM:



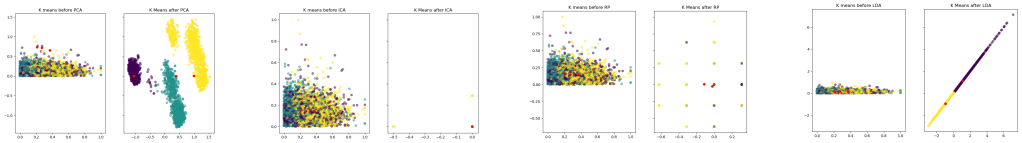
LDA was the best in terms of training time for both Kmeans and EM probably due to the fact that it reduces the data into one dimension. ICA was the slowest from all dimension reduction algorithms since it had to train for the independent components.

Dataset2

Kmeans:

Running Kmeans on the projected data

PCA **ICA** **RP** **LDA**



Comparison scores

Kmeans train					
Scores	**Original**	**After PCA**	**After ICA**	**After RP**	**After LDA**
Homogeneity	0.143	0.143	0	0.145	0.27
Completeness	0.073	0.073	0	0.119	0.216
V-measure	0.097	0.097	0	0.131	0.24
Rand Score	0.042	0.042	0.005	0.179	0.235
Mutual Info	0.096	0.096	0	0.131	0.24

Kmeans test					
Scores	**Original**	**After PCA**	**After ICA**	**After RP**	**After LDA**
Homogeneity	0.143	0.143	0	0.141	0.277
Completeness	0.073	0.073	0.001	0.115	0.221
V-measure	0.096	0.096	0.001	0.127	0.246
Rand Score	0.04	0.04	-0.011	0.172	0.225
Mutual Info	0.095	0.095	-0.001	0.126	0.246

Number of clusters: Original-3, PCA-3, ICA-3, RP-2 and LDA-1

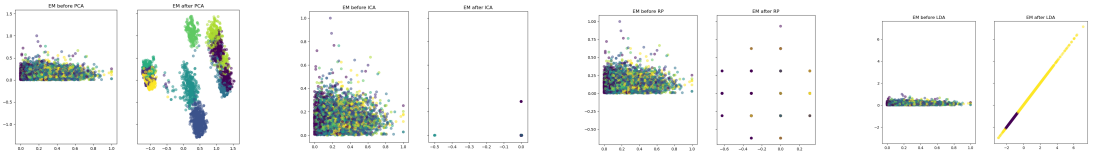
The number of clusters was chosen again for each dimension reduction algorithms by calculating the WCSS score. The value of the clusters obtained was same as

original which are 2. PCA and LDA were able to cluster the data very well into individual clusters as shown in the graphs above with centroids clearly clustering the data, but RP and ICA did not perform very well for the clustering in this case. The comparison scores for the ICA were also bad when comparing with original labels showing the features are not mutually dependent. Since PCA performs well it shows there is a high co-relation among the features.

EM:

Running EM on the projected data

PCA **ICA** **RP** **LDA**



Comparison scores

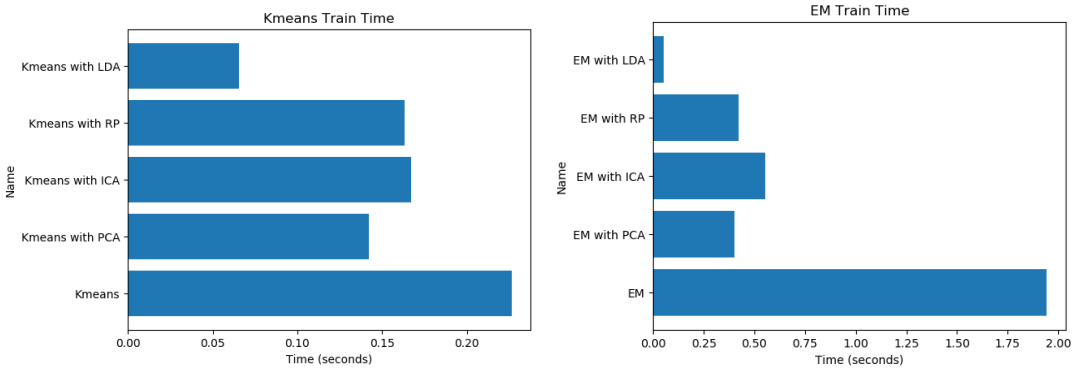
EM train					
Scores	**Original**	**After PCA**	**After ICA**	**After RP**	**After LDA**
Homogeneity	0.22	0.244	0.004	0.173	0.193
Completeness	0.058	0.063	0.004	0.094	0.163
V-measure	0.092	0.101	0.004	0.122	0.177
Rand Score	0.031	0.041	-0.026	0.141	0.013
Mutual Info	0.091	0.1	0.004	0.121	0.177

EM test					
Scores	**Original**	**After PCA**	**After ICA**	**After RP**	**After LDA**
Homogeneity	0.233	0.248	0.002	0.16	0.18
Completeness	0.06	0.064	0.002	0.088	0.153
V-measure	0.096	0.102	0.002	0.113	0.165
Rand Score	0.031	0.041	-0.019	0.134	0.004
Mutual Info	0.094	0.1	0.001	0.113	0.165

Number of clusters: Original-9, PCA-9, ICA-3, RP-3 and LDA-1

The number of clusters was chosen again for each dimension reduction algorithms by calculating the Silhouette score. The value of the clusters obtained was same as original which are 2. The above comparison scores states how various dimension reduction algorithms lined up with the original labels which shows LDA did the best but the labels were not that closely related with the clusters.

Training time comparison between Kmeans and EM:



LDA was the best in terms of training time for both Kmeans and EM probably due to the fact that it reduces the data into one dimension. ICA was the slowest from all dimension reduction algorithms since it had to train for the independent

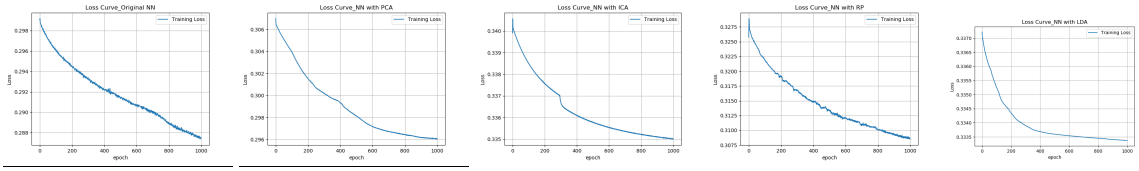
components. The original clustering for Kmeans and EM algorithm takes the maximum amount of training time, which proves dimension reduction is useful for reducing the training time.

Section4 – Running neural network for the new projected data

Neural network was re-ran on the projected data from PCA, ICA, RP and LDA on the second dataset (Adult income label) and accuracy score was calculated against the original test data. The parameters used for training the neural network was 5 hidden layers and 2 hidden nodes, which were obtained by hyper-parameter tuning against the learning curves. The below loss and accuracy curves were plotted for 1000 epochs:

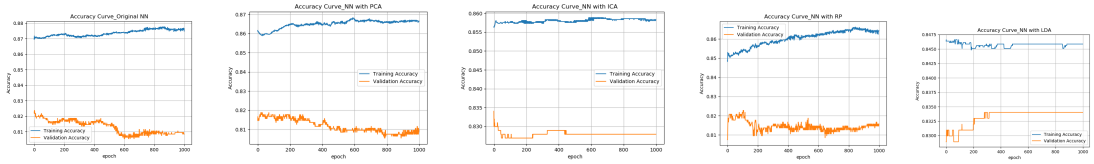
Loss curves

Original **PCA** **ICA** **RP** **LDA**



Accuracy curves

Original **PCA** **ICA** **RP** **LDA**



Accuracy score and Query time

	Accuracy score
Original	0.838630807
After PCA	0.834963325
After ICA	0.842909535
After RP	0.829462103
After LDA	0.841075795

	Query time
Original	0.000648975
After PCA	0.000771046
After ICA	0.000735044
After RP	0.000672102
After LDA	0.000400305

LDA had the good accuracy and also took the least query time. ICA had the best accuracy but took more query time. The high accuracy scores for ICA and LDA can be attributed to the fact that the data was more linear in nature since both ICA and LDA does the linear transformation. LDA also performs better in terms of query time since the number of dimensions is reduced to only one accounting for neural network to look for fewer features.

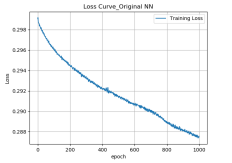
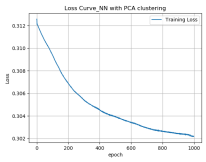
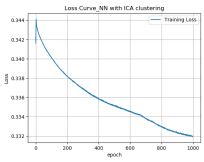
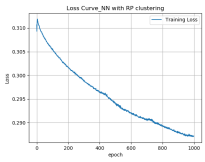
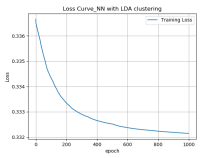
Section5 – Running neural network for the new projected data adding cluster labels as new features

Neural network was re-ran on the projected data from PCA, ICA, RP and LDA by adding the clustering labels as the new feature on the second dataset (Adult income

label) and accuracy score was calculated against the original test data. The parameters used for training the neural network was 5 hidden layers and 2 hidden nodes, which were obtained by hyper-parameter tuning against the learning curves. The below loss and accuracy curves were plotted for 1000 epochs:

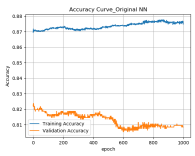
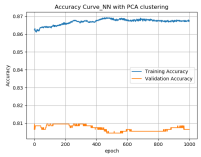
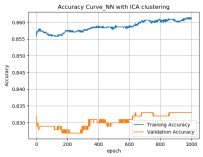
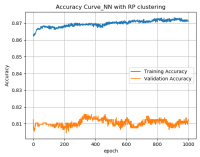
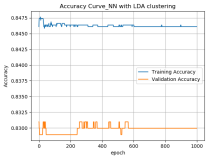
Loss curves

Original **PCA** **ICA** **RP** **LDA**

Accuracy curves

Original **PCA** **ICA** **RP** **LDA**

Accuracy score and Query time

	Accuracy score
Original	0.838630807
After PCA	0.822127139
After ICA	0.843520782
After RP	0.835574572
After LDA	0.832518337

	Query time
Original	0.000648975
After PCA	0.000494957
After ICA	0.000869036
After RP	0.00082612
After LDA	0.000385046

ICA performed the best in terms of accuracy score where as LDA took the least amount of query time. The high accuracy scores for ICA can be attributed to the fact that the data was more linear in nature since both ICA and LDA does the linear transformation. LDA also performs better in terms of query time since the number of dimensions is reduced to only one accounting for neural network to look for fewer features.

Citation:

[1] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html#sklearn.cluster.KMeans
[2] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.FastICA.html
[3] https://scikit-learn.org/stable/modules/random_projection.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html
[5] https://www.edupristine.com/blog/beyond-k-means
[6] https://www.linkedin.com/pulse/finding-optimal-number-clusters-k-means-through-elbow-asanka-perera
[7] Georgia Tech CS 7641 lecture videos and Piazza forum