Unsupervised Learning and Dimensionality Reduction

CS 7641: Machine Learning Assignment 3

By Divya Paduvalli (GT ID: dpaduvalli3/902913716)

Abstract

In this report, unsupervised learning and dimensionality reduction techniques were used to process two datasets and train it on clustering and neural network models. This report has been divided into five sections, each section describes an experiment. In the first section, K-Means (KM) and Expectation Maximization (EM) clustering models were applied on the datasets. In the second section, Principle Component Analysis (PCA), Independent Component Analysis (ICA), Randomized Projection (RP), and Linear Discriminant Analysis (LDA) were applied on the datasets. In the third section, clustering models were reapplied to the new datasets generated from the second section. In the fourth section, a neural network (NN) model was applied on the new datasets. In the fifth section, clustering models were used as dimensionality reduction models to generate new datasets and the NN model was reapplied. The analysis results presented in this report was accomplished using Python and Weka.

About the Data Sets

The datasets were downloaded from the UCI Machine Learning Repository. Please refer to the links provided in the README.txt to learn more about the attributes. The data sets used in this analysis were the following:

1. Default of Credit Card Clients: This data set was from a Taiwan-based credit card issuer whose customers had default credit card payments. There are 24 attributes, 30,000 instances, and 2 classes. The task was to predict the likelihood of default based on attributes such as past credit history, age, marital status etc.

2. Adult Census Income: This data set was an extraction from the 1994 census database. There are 14 attributes, 48,861 instances, and 2 classes. The task was to predict whether the income of an adult US citizen is either less than or greater than or equal to $50,000 per year based on attributes such as age, race, education, gender etc.

Why are the Classification Problems Interesting?

In the last few years, credit card issuers have become one of the major consumer lending products in the United States. This industry represents around 30% of total consumer lending (1). In a well-developed financial system, risk prediction is essential for predicting business performance and individual customer's credit risk. Credit card defaults not only result in significant financial losses to the bank but also damaged credit rating to the customer. Machine learning can be used in accurately predicting which customers are most probable to default and help notify customers who are on the verge of default. This in turn helps in reducing damage and uncertainty.

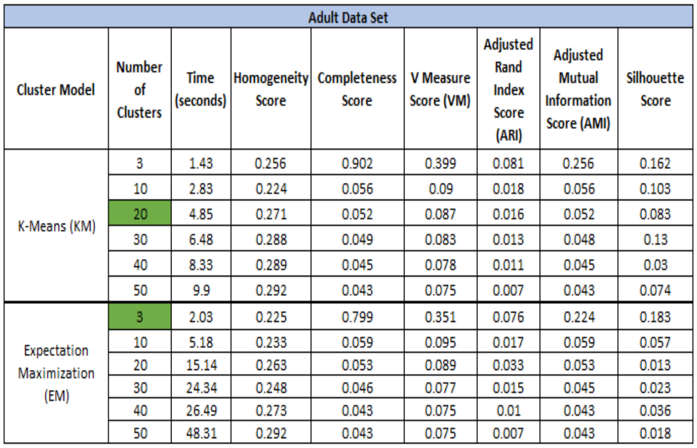
Census data is used by a wide variety of government agencies, businesses, academics, researchers etc. for a variety of purposes. The ability to predict a person's income using machine learning can be useful in applications like recommending housing, marketing of consumer or financial products, help business find locations to set up based on income, help the provision of healthcare plans and education, help make policy changes to increase household income in impoverished neighborhoods etc.

The datasets were selected mainly for their practical implications and for the results they gave when machine learning algorithms were applied. Both the data sets are non-trivial classification problems with more than 10,000 instances. This provides an opportunity to experiment with these datasets by applying different machine learning algorithms. Both the data sets have been resistant to good performance, since none of the algorithms have yielded exceptionally good results. One possible reason for this could be due to the nature of the datasets having imbalanced output classes and noise. However, in a lot of real-world application problems, data is not always perfect, and class imbalance issues are expected. It is important to note that both the datasets have not been artificially synthesized and are a direct reflection (or a subset) of real-world classification problems.

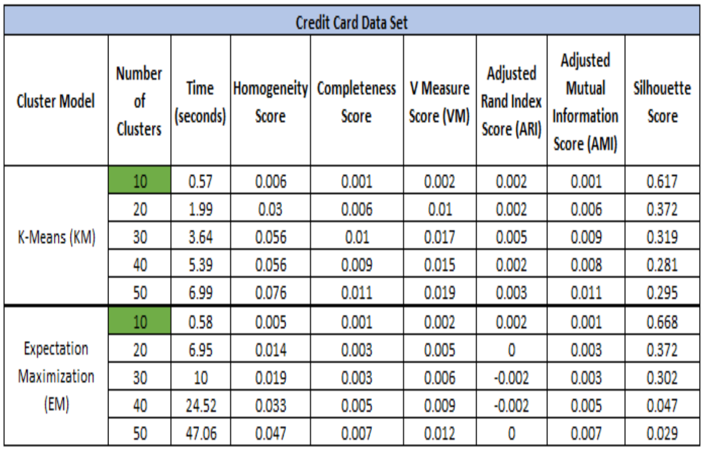
Preliminary Data Pre-Processing

Both the datasets contained rows with missing values which were removed prior to splitting the data. Categorical values from both the datasets were changed to binary nominal dummy values (0 and 1) to make it easier for the algorithms to process the data. MinMaxScaler was applied to scale down continuous features and to ensure that all the features were given equal importance. The datasets were then split into 70% training and 30% testing. Weka's resample filter instance was used to split the data. After splitting, the rows in the data sets were randomized to ensure there was no bias. The machine learning algorithms were applied to the training set first and then to the testing set.

Experiment 1: Run the Clustering Algorithms on the Datasets

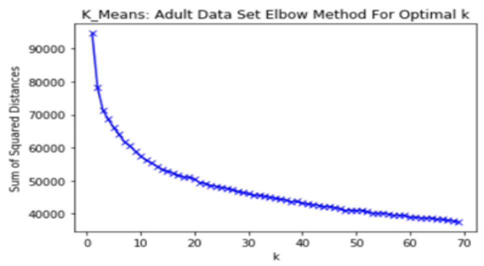


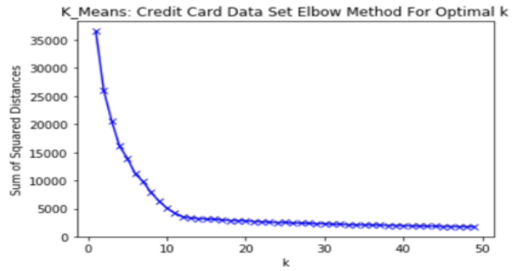
Adult Data Set								
Cluster Model	Number of Clusters	Time (seconds)	Homogeneity Score	Completeness Score	V Measure Score (VM)	Adjusted Rand Index Score (ARI)	Adjusted Mutual Information Score (AMI)	Silhouette Score
K-Means (KM)	3	1.43	0.256	0.902	0.399	0.081	0.256	0.162
	10	2.83	0.224	0.056	0.09	0.018	0.056	0.103
	20	4.85	0.271	0.052	0.087	0.016	0.052	0.083
	30	6.48	0.288	0.049	0.083	0.013	0.048	0.13
	40	8.33	0.289	0.045	0.078	0.011	0.045	0.03
	50	9.9	0.292	0.043	0.075	0.007	0.043	0.074
Expectation Maximization (EM)	3	2.03	0.225	0.799	0.351	0.076	0.224	0.183
	10	5.18	0.233	0.059	0.095	0.017	0.059	0.057
	20	15.14	0.263	0.053	0.089	0.033	0.053	0.013
	30	24.34	0.248	0.046	0.077	0.015	0.045	0.023
	40	26.49	0.273	0.043	0.075	0.01	0.043	0.036
	50	48.31	0.292	0.043	0.075	0.007	0.043	0.018

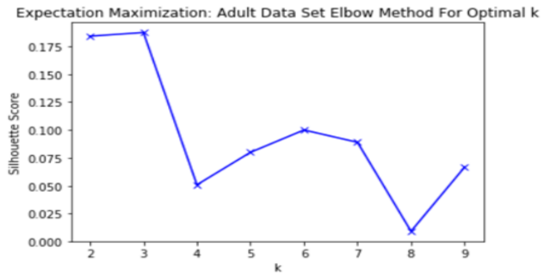


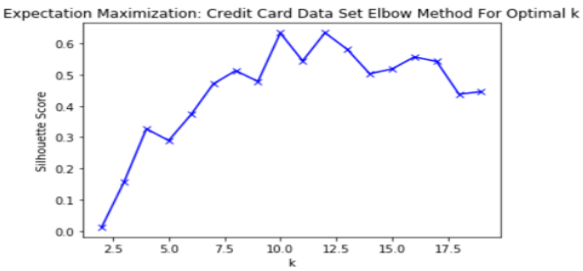
Credit Card Data Set								
Cluster Model	Number of Clusters	Time (seconds)	Homogeneity Score	Completeness Score	V Measure Score (VM)	Adjusted Rand Index Score (ARI)	Adjusted Mutual Information Score (AMI)	Silhouette Score
K-Means (KM)	10	0.57	0.006	0.001	0.002	0.002	0.001	0.617
	20	1.99	0.03	0.006	0.01	0.002	0.006	0.372
	30	3.64	0.056	0.01	0.017	0.005	0.009	0.319
	40	5.39	0.056	0.009	0.015	0.002	0.008	0.281
	50	6.99	0.076	0.011	0.019	0.003	0.011	0.295
Expectation Maximization (EM)	10	0.58	0.005	0.001	0.002	0.002	0.001	0.668
	20	6.95	0.014	0.003	0.005	0	0.003	0.372
	30	10	0.019	0.003	0.006	-0.002	0.003	0.302
	40	24.52	0.033	0.005	0.009	-0.002	0.005	0.047
	50	47.06	0.047	0.007	0.012	0	0.007	0.029

Cells highlighted in green indicates optimal k clusters







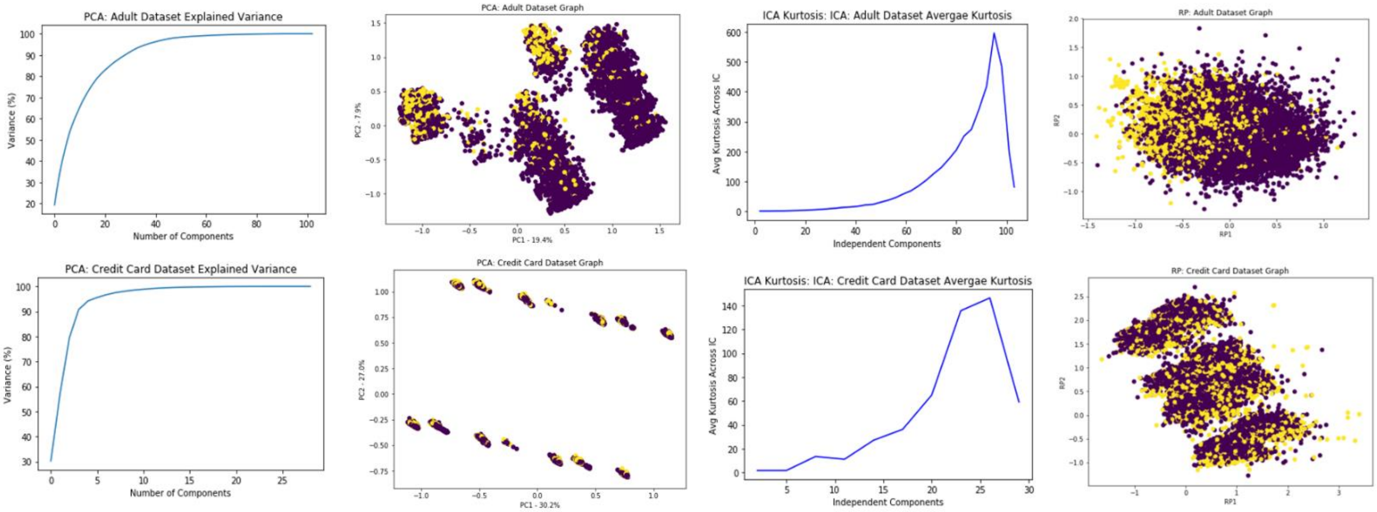


Typically, in unsupervised learning, we try to understand the inherent structure of our data without using explicitly-provided labels. A classic example would be a facial recognition technology that uses pixel information to predict an individual's identity. Technologies like these use clustering algorithms. In clustering, data points are divided into several groups such that similar data points are combined in groups or clusters. Two clustering models, K-Means (KM) and Expectation Maximization (EM), will be analyzed in this section.

In the KM model, each data point is iteratively assigned to one of the k clusters based mainly on how similar the features of the data points are. The KM algorithm starts with k random centroids which are used to assign data points to its closest cluster. The average of all the data points in a cluster is used to re-update the position of the centroids. This process is repeated several times until the values of the centroids become stable and the final stabilized centroids will be used to produce clusters. Similarly, the EM algorithm calculates probabilities of cluster memberships based on a mixture of different probability distributions and thus the end goal is to maximize the overall probability of the data given the distributions or clusters. The EM algorithm iteratively computes the mean and the standard deviation of each cluster. Unlike the KM model, which tries to assign data points to clusters to maximize the distances between the clusters, the EM model tries to compute probability of whether a data point belongs to a certain cluster.

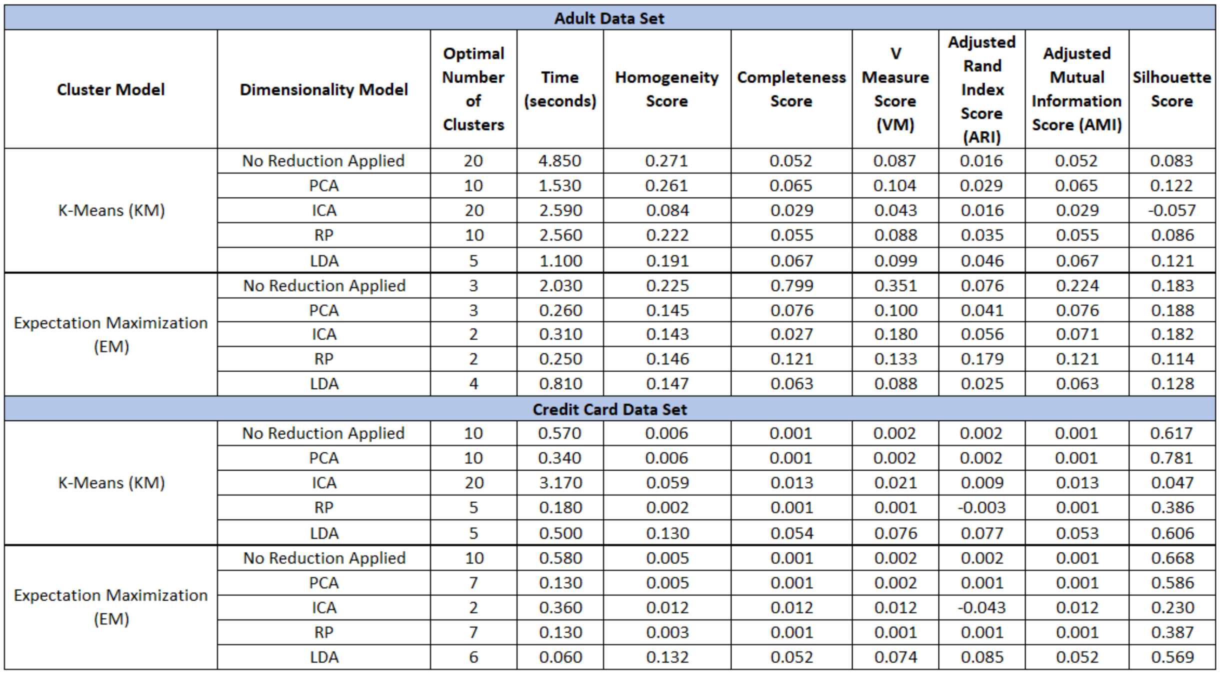
In this experiment, different metrics were used to analyze and evaluate the two clustering models. For the KM model, the elbow method was used to choose the optimal k which was 20 for the Adult dataset and 10 for the Credit Card dataset. The elbow method takes the total within cluster sum of squared distance (SSD) as a function of the total number of clusters. As k increases, SSD moves towards zero and thus the optimal k is selected such that increasing k does not significantly decrease the SSD. For the EM model, silhouette scores were used to find the optimal k which was 3 for the Adult data set and 10 for the Credit Card dataset. The silhouette score checks whether the clusters are well separated and compact by measuring the mean distance between the data points within the same cluster and the mean distance between the data points from one cluster to another cluster. If this score is closer to 1 then, then that clustering is better or optimal. I also experimented with Jensen-Shannon (JS) metric for the EM model. The JS metric measures similarity between two probability distributions. The lesser the JS distance between two clusters, the better the data fits the model. The problem that I encountered with this approach was, it was computationally very expensive to run and took a total of 30 minutes to create a simple graph for 20 clusters and the graph was extremely distorted since it had a lot of peaks and valleys which made it hard to pin point the optimal k value. In general, the overall computational time increased linearly as the number of clusters increased. Homogeneity and completeness scores were metrics used to measure how well the clustering models segregates data points belonging to different classes. Both the scores range from 0 to 1. If the scores are closer to 1, it indicates that the algorithm has done a fairly good job in predicting the right label for a datapoint. If you notice in the Adult data set, for both the models, as the homogeneity score increases the completeness score decreases. A reason for this behavior is because in homogeneity you can have multiple clusters where each cluster contains data points of a single class, but this is far from completeness which requires all data points of a specific class to be clubbed in a single cluster. Both the scores were very low for the Credit Card data set. The VM metric is an entropy-based metric which measures the harmonic mean between the homogeneity and completeness scores. The VM scores are much better when compared to the completeness scores. The ARI metric computes similarity measure between clusters (score range between -1 to 1) and AMI accounts for chance of identical clusters (score range between 0 to 1). These scores for the best k clusters highlighted in green seems to indicate that there are a lot of non-identical clustering labels and a possible reason for this behavior may be because the ground truth clustering is severely unbalanced.

Experiment 2: Apply Dimensionality Reduction Techniques on the Datasets



Principal Component Analysis (PCA) was used to reduce many correlated variables into a smaller number of uncorrelated variables or principle components. Python's PCA function was used. Some of these irrelevant correlated variables can decrease the performance of the machine learning algorithms which causes overfitting issues. Typically, the first few principle components explains most of the variance and eventually decreases as the number of components increase. From the above PC1 vs PC2 graphs, PC1 explains 19.4% of variation compared to PC2 which explains 7.9% for the Adult dataset. In the Credit Card dataset, 30% of variation is explained by PC1 and 27% is explained by PC2. In order to find the optimal number of components, I created a graph to plot the number of components vs explained variance for each variable. Based on the explained variance graphs, for the Adult data set, the optimal number of components was selected to be 40. 40 components might still seem very high, but it is important to note that the initial number of attributes was around 103. For the Credit Card dataset, number of components was selected to be just 5 (initial number of attributes was 29). Unlike PCA, Independent Component Analysis (ICA), tried to maximize independence by finding a linear transformation of the feature space into a new feature space such that each of the individual new features are mutually and statistically independent of one another and their mutual information was zero. Python's FastICA function was used. The optimal number of components for ICA was chosen by inspecting the kurtosis which is a measure of whether the data sets have light (high kurtosis) or heavy tails (low kurtosis) relative to a normal distribution. Since ICA model is based on kurtosis maximization, for the Adult dataset, the number of components was selected to be 95 and for the Credit Card dataset it was 25. In Randomized Projection (RP), dimensionality of the data was reduced by trading a certain amount of accuracy for smaller model sizes and faster processing times. I used Weka's RandomProjection function and used J48 classifier, which measures components based on accuracy score, to select the optimum number of components. For the Adult dataset 30 components were selected and 5 for the Credit Card dataset. I have selected Linear Discriminant Analysis (LDA) as my choice for doing dimensionality reduction. In LDA, the number of dimensions was reduced by projecting the data to a linear subspace which maximize the separation between the output classes while retaining information. The number of components in this case is the number of linear discriminants that should be kept. The number of components was determined by the total number of output labels and for both the data sets, which was 2, and hence 1 (i.e. number of classes - 1) component was selected for both the datasets. I verified this by creating a function in python to determine how much variance is explained by every component and it turned out that just a single component was enough to explain 95% of the variation.

Experiment 3: Reproduce Cluster Experiments on Dimensionality Reduced Datasets



Adult Data Set									
Cluster Model	Dimensionality Model	Optimal Number of Clusters	Time (seconds)	Homogeneity Score	Completeness Score	V Measure Score (VM)	Adjusted Rand Index Score (ARI)	Adjusted Mutual Information Score (AMI)	Silhouette Score
K-Means (KM)	No Reduction Applied	20	4.850	0.271	0.052	0.087	0.016	0.052	0.083
	PCA	10	1.530	0.261	0.065	0.104	0.029	0.065	0.122
	ICA	20	2.590	0.084	0.029	0.043	0.016	0.029	-0.057
	RP	10	2.560	0.222	0.055	0.088	0.035	0.055	0.086
	LDA	5	1.100	0.191	0.067	0.099	0.046	0.067	0.121
Expectation Maximization (EM)	No Reduction Applied	3	2.030	0.225	0.799	0.351	0.076	0.224	0.183
	PCA	3	0.260	0.145	0.076	0.100	0.041	0.076	0.188
	ICA	2	0.310	0.143	0.027	0.180	0.056	0.071	0.182
	RP	2	0.250	0.146	0.121	0.133	0.179	0.121	0.114
	LDA	4	0.810	0.147	0.063	0.088	0.025	0.063	0.128
Credit Card Data Set									
K-Means (KM)	No Reduction Applied	10	0.570	0.006	0.001	0.002	0.002	0.001	0.617
	PCA	10	0.340	0.006	0.001	0.002	0.002	0.001	0.781
	ICA	20	3.170	0.059	0.013	0.021	0.009	0.013	0.047
	RP	5	0.180	0.002	0.001	0.001	-0.003	0.001	0.386
	LDA	5	0.500	0.130	0.054	0.076	0.077	0.053	0.606
Expectation Maximization (EM)	No Reduction Applied	10	0.580	0.005	0.001	0.002	0.002	0.001	0.668
	PCA	7	0.130	0.005	0.001	0.002	0.001	0.001	0.586
	ICA	2	0.360	0.012	0.012	0.012	-0.043	0.012	0.230
	RP	7	0.130	0.003	0.001	0.001	0.001	0.001	0.387
	LDA	6	0.060	0.132	0.052	0.074	0.085	0.052	0.569

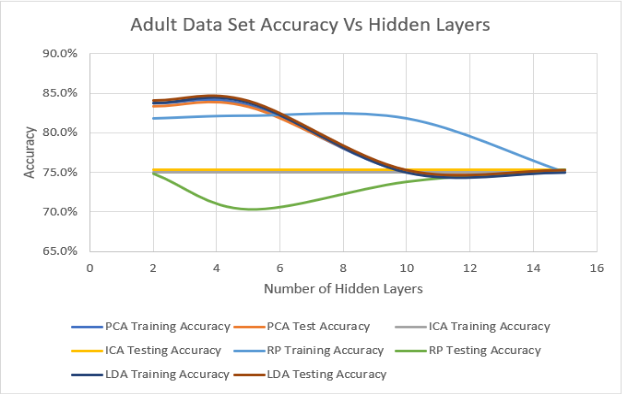
Similar to experiment 1, SSD vs k was used for the KM model and silhouette score vs k was used for the EM model to determine the number of clusters to use for each of the clustering algorithms that used specific dimensionality reduced datasets. For both the datasets, it was interesting to notice that ICA gave the highest number of optimal clusters for the KM model and the lowest for the EM model. The reason for this behavior could be because of the higher complexity involved due to the combination of the number of components used by the algorithm and the size of the dataset. ICA is the only technique that has used the highest number of components for both the datasets (95 for the Adult dataset and 25 for the Credit Card dataset). Due to this, the clustering models require a lot more clusters in correctly grouping the datapoints. LDA data had the lowest optimal clusters specifically for the KM model mainly due to just using 1 component which significantly reduces complexity thus requiring less clusters. It was interesting to see that the number of clusters for both the clustering models, formed after using PCA data, was not significantly different as compared to the original datasets. I was expecting the clustering models to significantly reduce the clusters for PCA since PCA had reduced the number of components for both the datasets significantly. A possible reason for this is because the number of clusters is also dependent on the size of the dataset and thus the difference isn’t significant despite the reduced features. RP was ran multiple times (20 times) and the overall accuracy and evaluation metrics did not vary much (~1%). In general, computational time improved significantly as compared to experiment 1 and the improvement was linear. The evaluation metrics performance has not been significantly higher as compared to experiment 1, which indicates that the clustering algorithms has done a decent job in clustering the labels.

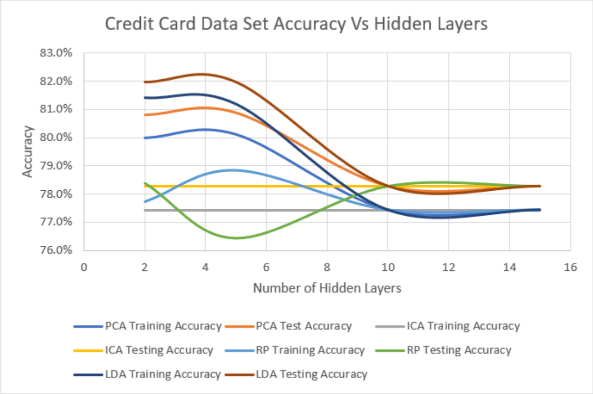
Experiment 4: Reproduce Neural Network (NN) Models on Dimensionality Reduced Datasets

Adult Data Set (Learning Rate = 0.1, Momentum = 0.1, Hidden Layers = 1, Hidden Nodes = 4)				
Datset Used	Model Build Time (sec)	Model Test Time (sec)	Training Accuracy	Test Accuracy
Original	4.868	0.030	83.835%	83.136%
PCA	4.065	0.012	83.697%	83.545%
ICA	4.593	0.060	75.020%	75.312%
RP	3.560	0.003	81.940%	74.914%
LDA	2.060	0.001	83.801%	84.087%

Credit Card Data Set (Learning Rate = 0.1, Momentum = 0.1, Hidden Layers = 1, Hidden Nodes = 4)				
Datset Used	Model Build Time (sec)	Model Test Time (sec)	Training Accuracy	Test Accuracy
Original	3.510	0.020	81.057%	81.736%
PCA	3.246	0.030	79.773%	80.723%
ICA	3.560	0.040	77.432%	78.279%
RP	3.210	0.020	77.688%	78.392%
LDA	2.040	0.010	81.274%	81.950%

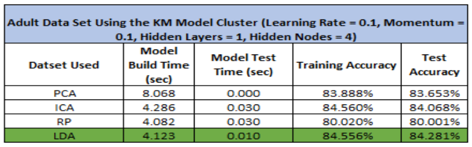
Cells highlighted in green indicates high testing set accuracy





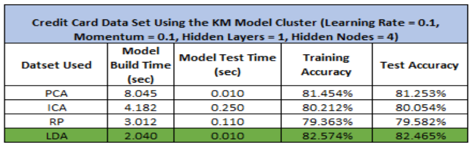
MLPClassifier function was used in python to evaluate the performance and behavior of the neural networks (NN). In the NN algorithm, four parameters were tuned namely, learning rate, momentum, hidden layers, and hidden nodes per layer. Learning rate is how much weight is assigned to new examples. The momentum coefficient adds speed to the training of any multi-layer perceptron by adding part of the already occurred weight changes to the current weight change. Hidden layer is the layer between input layers and output layers. Hidden nodes are the number of neurons that goes in each hidden layer. I experimented with different hidden nodes and found that anything beyond 4 nodes did not significantly improve the testing accuracy and hence kept 4. I conducted two experiments to understand how the accuracy changes with different datasets. In my first experiment for both the datasets, I set the learning rate to 0.1, momentum to 0.1, hidden layers to 1, and hidden nodes to 4. The reason why I chose these parameters is because I wanted to start off with low complexity and have consistent parameters for different datasets to see the differences in their accuracy. From the above table, it is interesting to notice that other than LDA, the testing performance for the ICA and RP datasets have generally been poor when compared to the original datasets. The highest testing accuracy for both the datasets is achieved by the LDA datasets and the second highest by the PCA datasets. A possible explanation for this behavior is because since the LDA datasets had the lowest number of components, complexity of neural network is reduced significantly and thus the accuracy increased a lot. The testing accuracy of the PCA model for both the datasets was very similar to the accuracy score of the original dataset. Typically, the linear transformation that the PCA model performs can also be performed the by the input layer weights of the NN model. In addition, NN also has a very strong non-linear computing power. So, using PCA (or any other techniques) is not necessary. Although, as the number of weights in the NN model increases the amount of data needed to accurately determine the weights also increases rapidly, and thus dimensionality reduction techniques can help to reduce the size of the network and the amount of data needed to train the model. In general, PCA, ICA, and RP does not take into account the discriminative feature information that distinguishes one class (target variable) from another and thus some of the useful features gets eliminated in this process. So there no guarantee that these techniques will result in improved performance in terms of accuracy as seen in the table above. In my second experiment, I changed the number of hidden layers and kept the hidden nodes constant. Accuracy generally decreased when hidden layers increased. In most practical cases it is advised not to use more than one hidden layer because the back-propagation algorithm become less effective and this is evident in the graphs above. This also causes overfitting problems where the network will perform very well on the training set but may perform poorly on the test set.

Experiment 5: Reproduce Neural Network (NN) Experiments on Newly Projected Data



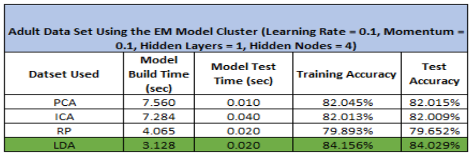
Adult Data Set Using the KM Model Cluster (Learning Rate = 0.1, Momentum = 0.1, Hidden Layers = 1, Hidden Nodes = 4)

Datset Used	Model Build Time (sec)	Model Test Time (sec)	Training Accuracy	Test Accuracy
PCA	8.068	0.000	83.888%	83.653%
ICA	4.286	0.030	84.560%	84.068%
RP	4.082	0.030	80.020%	80.001%
LDA	4.123	0.010	84.556%	84.281%



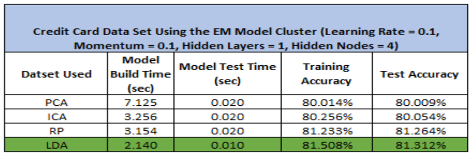
Credit Card Data Set Using the KM Model Cluster (Learning Rate = 0.1, Momentum = 0.1, Hidden Layers = 1, Hidden Nodes = 4)

Datset Used	Model Build Time (sec)	Model Test Time (sec)	Training Accuracy	Test Accuracy
PCA	8.045	0.010	81.454%	81.253%
ICA	4.182	0.250	80.212%	80.054%
RP	3.012	0.110	79.363%	79.582%
LDA	2.040	0.010	82.574%	82.465%



Adult Data Set Using the EM Model Cluster (Learning Rate = 0.1, Momentum = 0.1, Hidden Layers = 1, Hidden Nodes = 4)

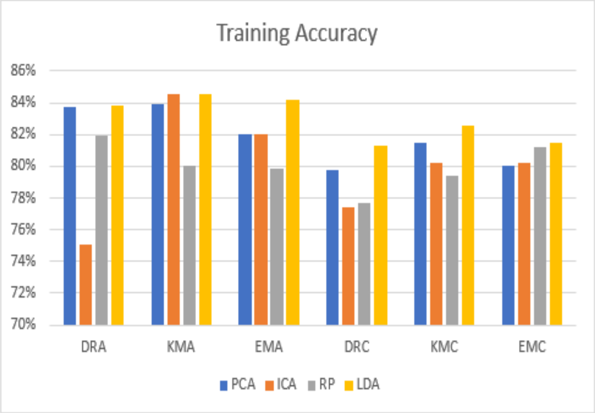
Datset Used	Model Build Time (sec)	Model Test Time (sec)	Training Accuracy	Test Accuracy
PCA	7.560	0.010	82.045%	82.015%
ICA	7.284	0.040	82.013%	82.009%
RP	4.065	0.020	79.893%	79.652%
LDA	3.128	0.020	84.156%	84.029%

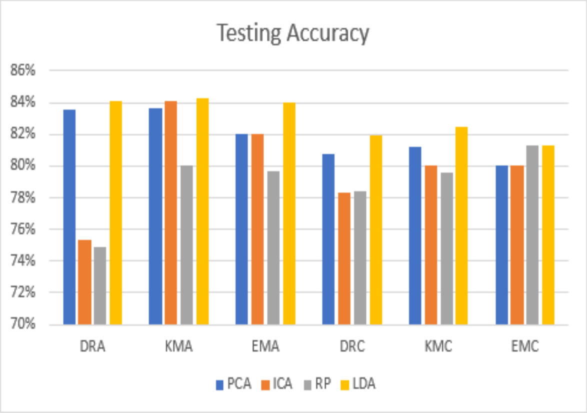


Credit Card Data Set Using the EM Model Cluster (Learning Rate = 0.1, Momentum = 0.1, Hidden Layers = 1, Hidden Nodes = 4)

Datset Used	Model Build Time (sec)	Model Test Time (sec)	Training Accuracy	Test Accuracy
PCA	7.125	0.020	80.014%	80.009%
ICA	3.256	0.020	80.256%	80.054%
RP	3.154	0.020	81.233%	81.264%
LDA	2.140	0.010	81.508%	81.312%

Cells highlighted in green indicates high testing set accuracy





Cluster Model	Dimensionality Model	Optimal Number of Clusters (k)	
		Adult Data Set	Credit Card Data set
K-Means (KM)	PCA	10	10
	ICA	20	20
	RP	10	5
	LDA	5	5
Expectation Maximization (EM)	PCA	3	7
	ICA	2	2
	RP	2	7
	LDA	4	6

The analysis in this section was performed using Python. The purpose of this section is to run the clustering algorithms on the dimensionality reduced datasets and include the clusters as features for the NN model. Just as in the previous section, I set the the parameters for the NN model as follows: learning rate = 0.1, momentum = 0.1, hidden layers = 1, and hidden nodes = 4. These parameters were chosen to reduce complexity and be consistent across different datasets. In experiment 2, when clustering experiments were reproduced on the dimensionality reduced datasets, the optimal number of clusters (indicated in the above table) were selected using the SSD and the silhouette scores method. These clustering results were added back to the dimensionality reduced datasets as a new feature addition. I created a total of 8 dimensionality reduced datasets, where 4 datasets belonged to the KM model and the rest 4 belonged to the EM model. The total number of dimensions obtained for the Adult and the Credit Card dataset after adding the cluster results were as follows: PCA = 41 & 6, ICA = 96 & 26, RP = 31& 6, and LDA = 2 & 2. In the above accuracy graphs, DRA = Dimensionality Reduced Adult dataset, KMA = Dimensionality Reduced K-Means Adult dataset, EMA = Expectation Maximization Dimensionality Reduced Adult dataset, DRC = Dimensionality Reduced Credit Card dataset, KMC = Dimensionality Reduced K-Means Credit Card dataset, and EMC = Expectation Maximization Dimensionality Reduced Credit Card dataset. Based on the above results, including cluster information did increase the performance of the NN models It is interesting to notice that, just like in experiment 4, LDA performed the best for both the datasets. In general, all the datasets performed better after clustering information was included. This indicates that including clustering information, makes the data linearly separable which in turn has a vital effect on the NN model's weights. In the lectures, we learnt that if the data is linearly separable, perceptron threshold rule can find the optimal weights that maps inputs to outputs and has a guarantee finite convergence and thus the accuracy score increased.

Conclusion

In this report, five experiments were conducted to understand how machine learning models would behave on data sets that have been processed differently. In the first experiment, clustering models were applied on non-transformed datasets and SSD or the elbow method and silhouette scores were used to find the optimal number of clusters. For the Adult dataset, 20 k clusters were selected for the KM model and 3 for the EM model. For the Credit Card dataset, 10 k clusters were selected for the KM model and 10 for the EM model. In the second experiment, the number of components selected for the Adult dataset includes, 40 for PCA, 95 for ICA, 30 for RP, and 1 for LDA. For the Credit Card dataset, 5 for PCA, 25 for ICA, 5 for RP, and 1 for LDA. In the third, fourth, and the fifth experiments, clustering and NN models were applied on the transformed datasets and cluster values, training and testing time, training and testing accuracy scores were calculated.

Acronyms used in this Report

1. KM: K-Means
2. k: Number of Clusters
3. EM: Expectation Maximization
4. PCA: Principle Component Analysis
5. ICA: Independent Component Analysis
6. RP: Randomized Projection
7. LDA: Linear Discriminant Analysis
8. NN: Neural Network
9. SSD: Sum of Squared Distance
10. JS: Jensen-Shannon Metric
11. VM: V-Measure Metric
12. ARI: Adjusted Rand Index Metric
13. AMI: Adjusted Mutual Information Metric
14. DRA = Dimensionality Reduced Adult dataset
15. KMA = Dimensionality Reduced K-Means Adult dataset
16. EMA = Expectation Maximization Dimensionality Reduced Adult dataset
17. DRC = Dimensionality Reduced Credit Card dataset
18. KMC = Dimensionality Reduced K-Means Credit Card dataset
19. EMC = Expectation Maximization Dimensionality Reduced Credit Card dataset.

Sources

1. http://salserver.org.aalto.fi/opinnot/mat-2.4177/2018/McKinseyFinal.pdf
2. https://www.cs.waikato.ac.nz/ml/weka/
3. https://scikit-learn.org/stable/
4. Adult Data Set: http://archive.ics.uci.edu/ml/datasets/Adult
5. Credit Card Default Data Set: http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients