Unsupervised Learning and Dimensionality Reduction

Rui Hu; rhu63@gatech.edu

CS 7641 Machine Learning Assignment 3

1. Introduction

In this assignment, we explore the attributes of different clustering and dimensionality reduction algorithms. The purpose of the assignment is to: 1. Implement clustering algorithms and compare the results; 2. Implement dimensionality reduction algorithms and compare the results; 3. Combine clustering and dimensionality reduction algorithms to pre-process data to train neural networks. The two clustering algorithms tested are k-means and expectation maximization (EM). The four dimensionality reduction methods implemented are: principal component analysis (PCA), independent component analysis (ICA), random projection (RP) and Random Forest.

2. Datasets

2.1 Phishing Website Data Set

This dataset contains the important features that have proved to be sound and effective in predicting phishing websites. Since internet fraudulence is gaining popularity, this dataset can help security professionals to identify phishing website characters. The dataset has 30 features and 2456 instances. One-hot encoding is required for categorical features and the processed data has 46 features.

2.2 Default of credit card clients Data Set

This type of study is critical to banks and insurance companies as it will help them to make decisions on whether to increase interest rates. Machine learning technics can help predict default payments and let banks prepare beforehand. It has 30K instances with 24 features.

3. Clustering

We will test two clustering algorithms: K-means and Expectation Maximization (EM) on both datasets and compare their performance.

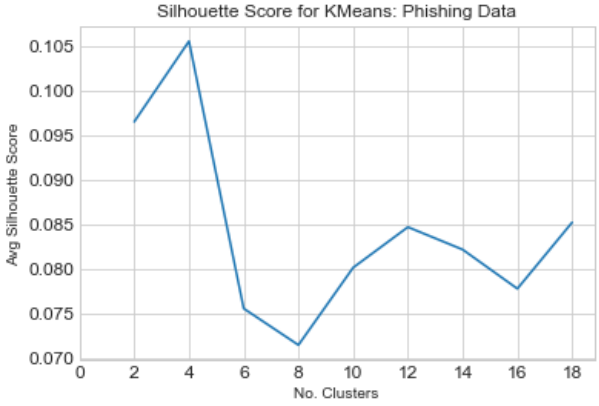
3.1 K-means

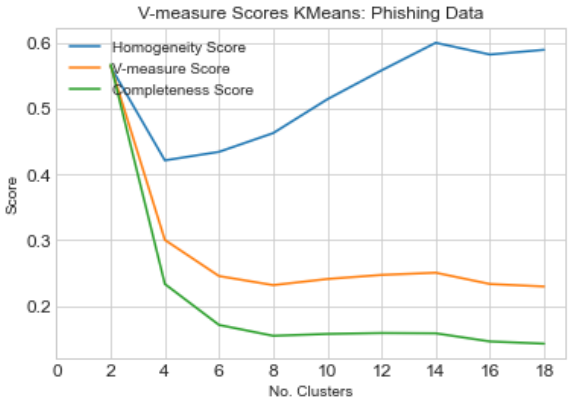
K-means algorithm is an iterative algorithm that tries to partition the dataset into K distinct clusters and minimizes within-cluster variances. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid is at the minimum [3]. Since k-means is susceptible to local optima if the initial cluster centers are not well located, we will report the average metrics over 5 models for each number of k clusters. For both datasets, we train the clustering algorithm on 70% of training set, pick the appropriate K value and then reports its accuracy and f1 score on the testing set.

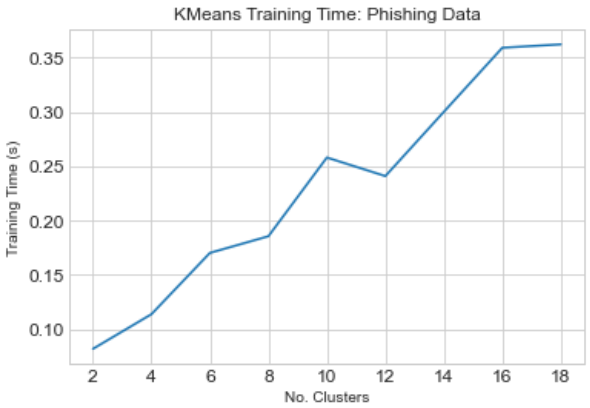
Several methods are used in this report to find the optimal value of k, such as silhouette coefficient and V-measure score. The silhouette coefficient is a measure of how similar a data point is within-cluster compared to other clusters. It ranges from -1 to 1. A score of 1 means that the data point i is very compact within the cluster to which it belongs and far away from the other clusters [4]. The V-measure is the harmonic mean between homogeneity and completeness, where homogeneity measures how much the sample in a cluster are similar and completeness measures how much similar samples are put together by the clustering algorithm [5].

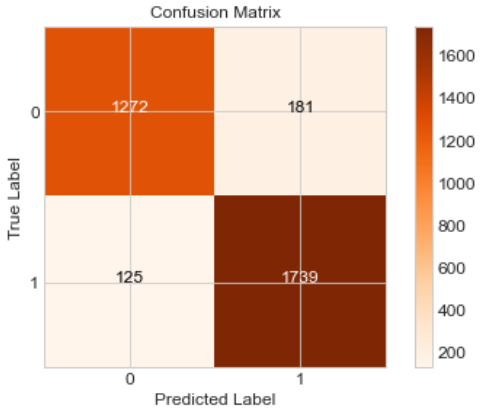
Dataset 1 (Phishing Websites)

For the phishing website data, we can see that both silhouette score and V-measure score are relatively high when K=2. Homogeneity score increases again when K=14, but since the V-measure score is low so we discard this K.









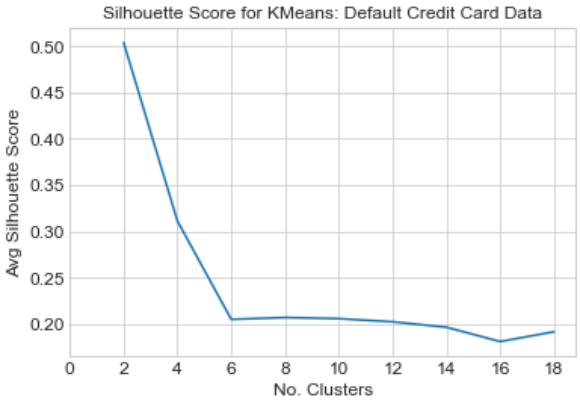
On the testing set, K=2 gives us accuracy of 0.9093 and F1 score of 0.9184. The other metrics are included in Table 1. The confusion matrix is shown above. Training time increases when No. of clusters increase, as expected.

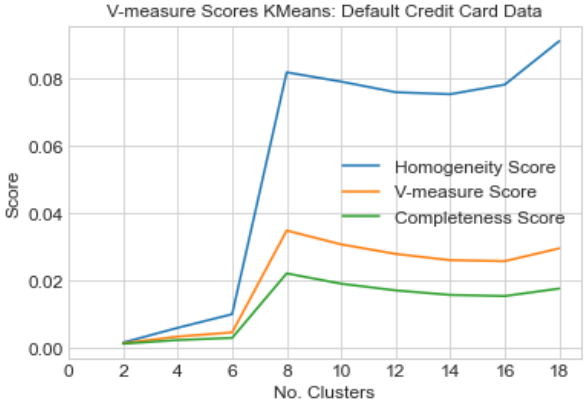
Data	K	training time	No. iter	f1 score	accuracy	precision	recall	AUC
Phishing Website	2	0.0486	10	0.9184	0.9093	0.9122	0.9246	0.9075
Default Credit Card	8	0.2419	31	0.3781	0.8048	0.6297	0.2701	0.6127

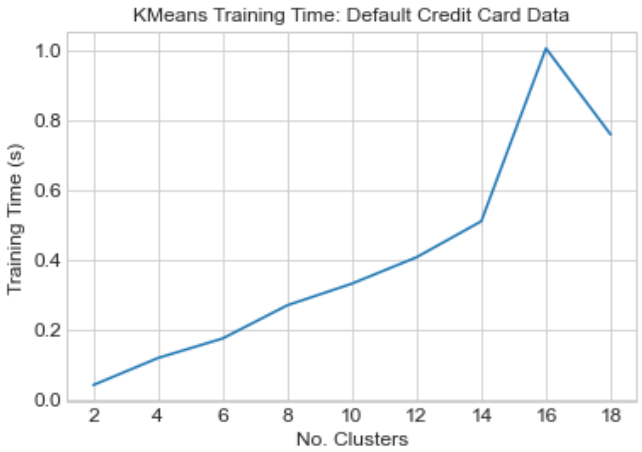
Table 1

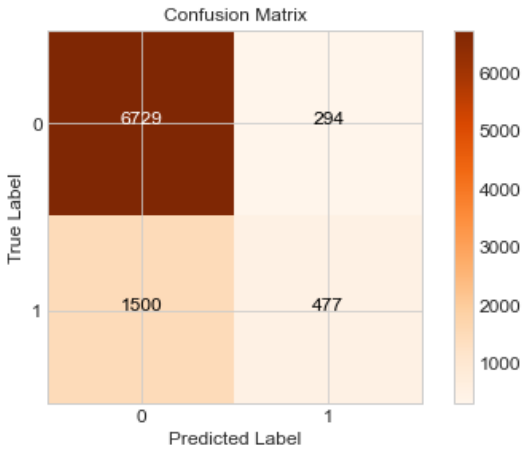
Dataset 2 (Default Credit Card)

For the Default Credit Card data, we can see that V-measure score increased significantly when K=8. It increased again when K=18, but we generally prefer lower values of K since we don't want our models to be too complex.









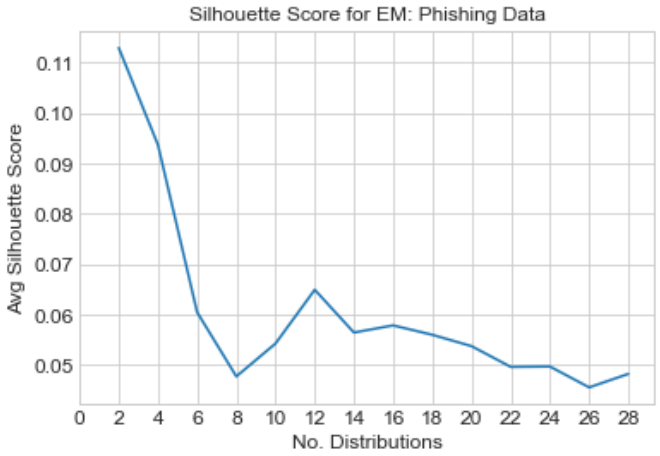
On the testing set, K=8 gives us accuracy of 0.8048 and F1 score of 0.3781. The other metrics are included in Table 1. The confusion matrix is shown above. We cans see that the model gives a lot of false negatives, thus a low recall score. Training time increases when No. of clusters increase, as expected.

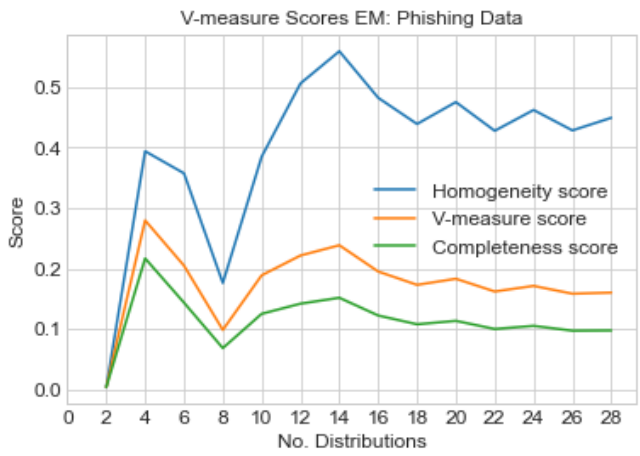
3.2 Expectation Maximization (EM)

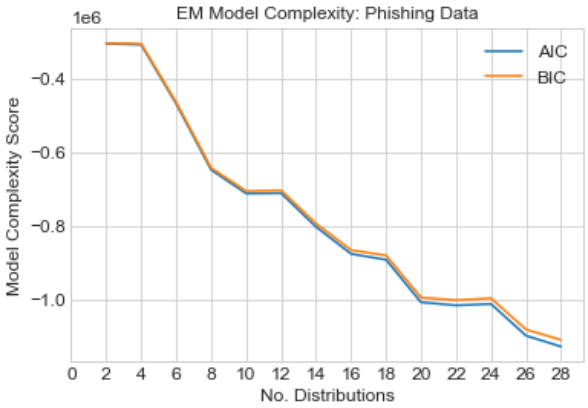
The Expectation-Maximization Algorithm is an approach for maximum likelihood estimation in the presence of latent variables. It's an iterative approach that cycles between two steps: E-step and M-step. The first step estimates the latent variables, called the estimation-step or E-step. The second step computes the maximum-likelihood estimators to update our parameter estimate, called the maximization-step or M-step. When we use Gaussian function to be our model estimation, this is called Gaussian Mixture Model. For details of EM please refer to [6]. Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) are also observed to control the model complexity [7]. A 70-30 training-testing split is used on both datasets. We train the clustering algorithm on 70% of training set, pick the appropriate number of distributions and then reports its accuracy and f1 score on the testing set.

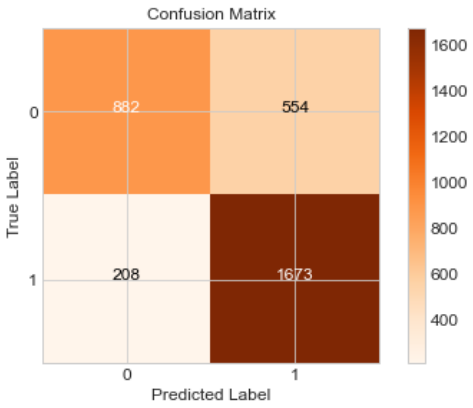
Dataset 1 (Phishing Websites)

For the phishing website data, we can see that V-measure score peaked when No. distribution=4. Homogeneity score increases again when No. distribution =14, but since the V-measure score is low so we discard it.









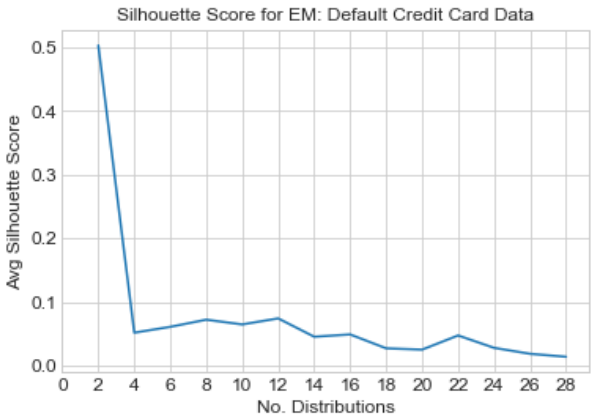
On testing set, 4 distributions give us accuracy of 0.7703 and F1 score of 0.8145. The other metrics are included in Table 2. The confusion matrix is shown above.

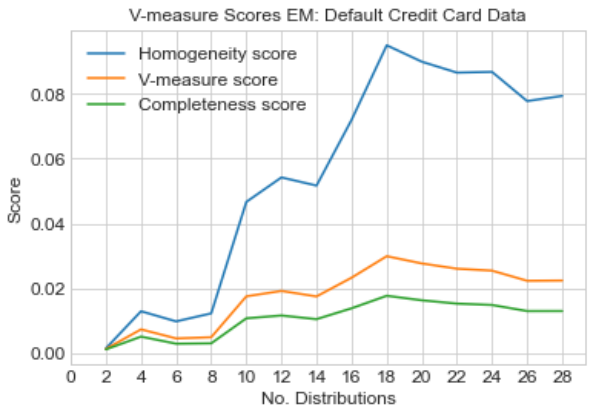
Data	No. Distributions	training time	No. iter	f1 score	accuracy	precision	recall	AUC
Phishing Data	4	0.1333	8	0.8145	0.7703	0.7512	0.8894	0.7518
Default Credit Data	12	0.4911	27	0.2694	0.7830	0.5172	0.1821	0.5671

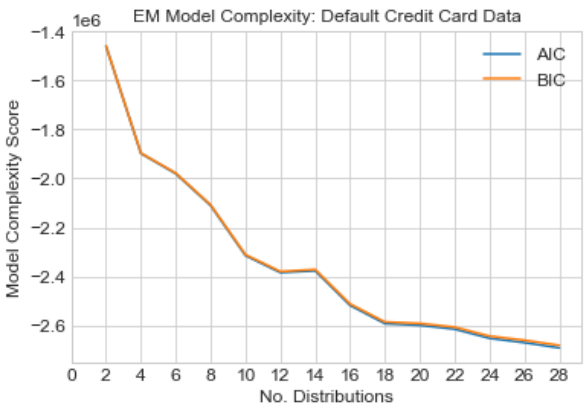
Table 2

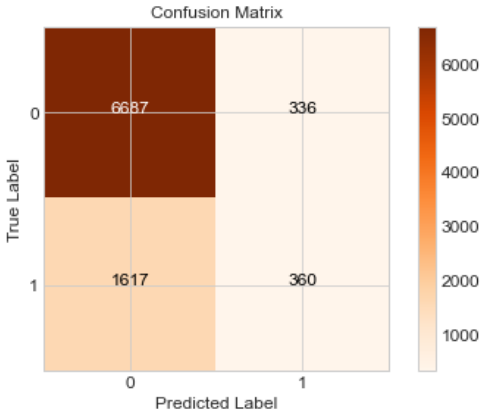
Dataset 2 (Default Credit Card)

For the Default Credit Card data, we can see that V-measure score increased significantly when number of distributions is 12. It increased again when No. distribution = 18, but the model is too complex at that point, so we discard it.









On testing set, No. distribution = 12 gives us accuracy of 0.7830 and F1 score of 0.3694. The other metrics are included in Table 2. The confusion matrix is shown above. We cans see that the clustering model again gives a lot of false negatives, meaning the data set is not easily separable.

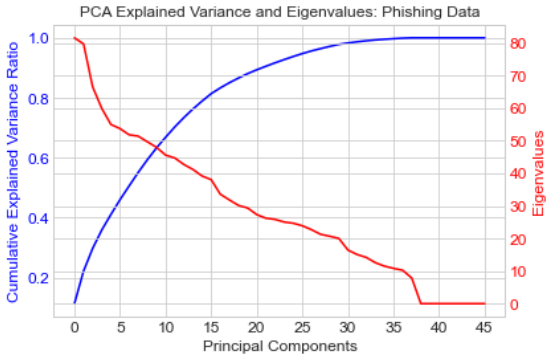
4. Dimensionality Reduction

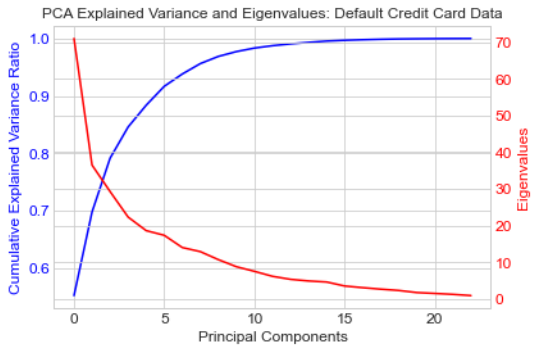
The main idea behind Dimensionality Reduction is to select or extract important features from datasets with many features. Feature selection works by finding the important features while feature extraction transforms features to lower dimensions. Several dimensionality reduction methods tested in this report, Principal Component Analysis (PCA), Independent Component Analysis (ICA), and Random Projection (RP), belong to feature extraction category; And Random Forest Classifier (RFC) is used as a feature selection method. In this experiment, a 50-50 training-testing split is used on both datasets, because RP requires too much RAM for a 70-30 split on the chosen datasets. We train the clustering algorithm on 50% of the samples, pick the appropriate number of distributions and then reports its accuracy and f1 score on the testing set.

4.1 Principal Component Analysis (PCA)

Principal component analysis is a technique for reducing the dimensionality of datasets with many features, increasing interpretability but at the same time minimizing information loss. This is accomplished by linearly transforming the data into a new coordinate system (principal component) where most of the variation in the data can be described with fewer dimensions than the initial data. [8]

After performing PCA on both datasets, we obtained the figures below. We can see that when the number of principal components increases, cumulative explained variance increased, and the Eigenvalues are decreasing. The original phishing website dataset has 46 features, but 25 principal components can already capture 95% of total variance. Default Credit Card Dataset originally has 24 features, while 10 principal components can capture more than 95% of total variance. PCA achieved better dimension reduction results on the default credit card dataset.

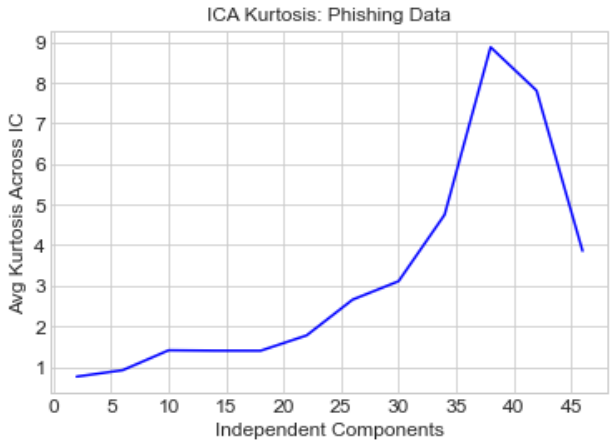


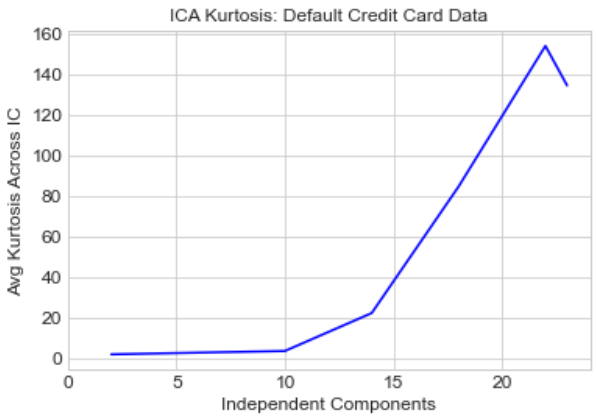


4.2 Independent Component Analysis (ICA)

Independent component analysis derives from signal processing. It is a computational method for separating a multivariate signal into additive subcomponents. It requires two assumptions: the source signals have non-Gaussian distributions, and the source signals are independent of each other. For details of ICA, please refer to [9].

Kurtosis is a measure that defines how heavily the tails of a distribution differ from the tails of a normal distribution. In other words, kurtosis identifies whether the tails of a given distribution contain extreme values. For ICA, higher kurtosis means better performance for that number of independent components. For phishing data, we should choose number of components to be 38, and for default credit card data, we should choose number of components to be 22.



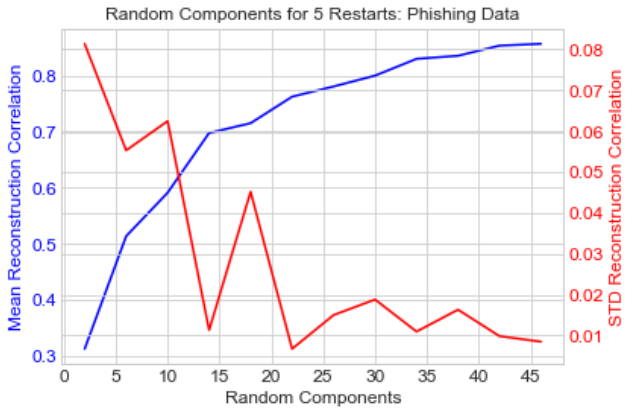


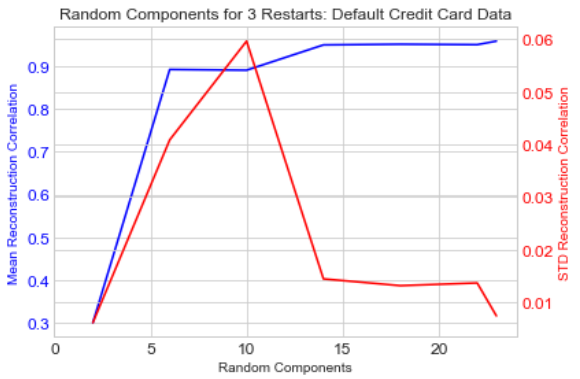
4.3 Random Projection (RP)

Random projection is based on Johnson-Lindenstrauss Lemma, which states that we can transform a high dimension data to a lower dimension data while nearly preserving the distance between any two points. It simply uses matrix multiplication to do transformation and thus is very fast. Time complexity = O(MND) where N,D are original dimension & new dimension respectively, and M is number of samples. Ideally, Random Projections requires a large N (1k-2k). The major steps in RP [10]:

1. Take dataset K, of the dimension Mx N (M=samples, N=original dimension/features)
2. Initialize a random 2d matrix R of size N x D where D= new reduced dimension
3. Normalize the columns of R making them unit length vectors.
4. Matrix multiplication J=K * R. J is the final output with dimension M x D.

For Phishing data, D=35 gives a relatively high reconstruction correlation and for Default credit data, D=15. Its dimension reduction power is worse than that of PCA.





4.4 Random Forest Classifier (RFC)

Random forest classifier can also be used for dimension reduction, feature selection, in this case. We use the impurity-based feature importance to rank all the features, and the top ones are selected. The importance of a feature is computed as the (normalized) total reduction of the Gini impurity brought by that feature.

11 features were selected from phishing data: 'SSLfinal_State_1', 'URL_of_Anchor_-1', 'SSLfinal_State_-1', 'URL_of_Anchor_1', 'SSLfinal_State_0', 'web_traffic_1', 'URL_of_Anchor_0', 'Prefix_Suffix', 'having_Sub_Domain_1', 'having_Sub_Domain_0', 'web_traffic_0'. The cumulative feature importance of the features is 0.941.

9 features were selected from default credit card data: 'PAY_0', 'PAY_2', 'PAY_3', 'PAY_4', 'PAY_5', 'PAY_6', 'PAY_AMT1', 'LIMIT_BAL', 'PAY_AMT3'. The cumulative feature importance of the features is 0.935.

4.5 Result Summary

In this experiment, we first performed all 4 dimensionality reduction methods on the training datasets, and picked the appropriate number of features, as is shown in the above section. Then, similar to section 3, we used clustering algorithms and selected the number of clusters. The overall results of 16 scenarios are as follows. Compared to ICA and RP, PCA and RFC are great at reducing the number of dimensions. For the two datasets, RFC kept only 24% (11/46) and 38% (9/24) of original features respectively; PCA kept 46% and 42% respectively. Moreover, PCA and RFC out-performed other methods by maintaining higher F1 score and accuracy in each dataset and clustering method combination.

Data	Dim Reduction	Clustering	No. Features remained	No. clusters	training time	f1 score	accuracy	precision	recall	AUC
Phishing Data	PCA	K-Means	25	2	0.0445	0.9243	0.9146	0.9129	0.9360	0.9119
	ICA	K-Means	38	8	0.1279	0.6463	0.6500	0.7391	0.5742	0.6597
	RP	K-Means	35	4	0.1232	0.7867	0.7433	0.7322	0.8500	0.7296
	RFC	K-Means	11	2	0.0261	0.9168	0.9068	0.9125	0.9211	0.9050
	PCA	EM	25	20	0.5371	0.8824	0.8573	0.8156	0.9610	0.8439
	ICA	EM	38	22	0.4080	0.7828	0.7576	0.7813	0.7843	0.7542
	RP	EM	35	10	0.3268	0.7492	0.7287	0.7720	0.7278	0.7288
	RFC	EM	11	2	0.0271	0.9179	0.9094	0.9268	0.9091	0.9094
Default Credit Data	PCA	K-Means	10	10	0.2640	0.3499	0.8015	0.6429	0.2403	0.6011
	ICA	K-Means	22	12	0.4502	0.3643	0.7901	0.5571	0.2706	0.6046
	RP	K-Means	15	10	0.3270	0.3650	0.7973	0.6003	0.2622	0.6062
	RFC	K-Means	9	12	0.2126	0.4711	0.8051	0.5934	0.3906	0.6571
	PCA	EM	10	12	1.8141	0.2763	0.7884	0.5755	0.1818	0.5718
	ICA	EM	22	12	1.6560	0.0000	0.7778	0.0000	0.0000	0.5000
	RP	EM	15	10	0.4528	0.0000	0.7778	0.0000	0.0000	0.5000
	RFC	EM	9	12	0.3982	0.2996	0.7980	0.6526	0.1944	0.5824

Table 3

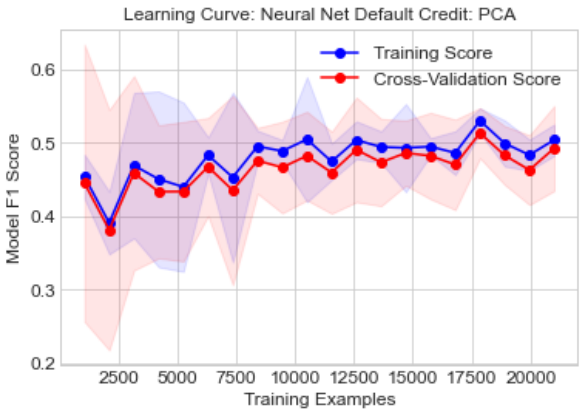
5. Dimensionality reduction and clustering

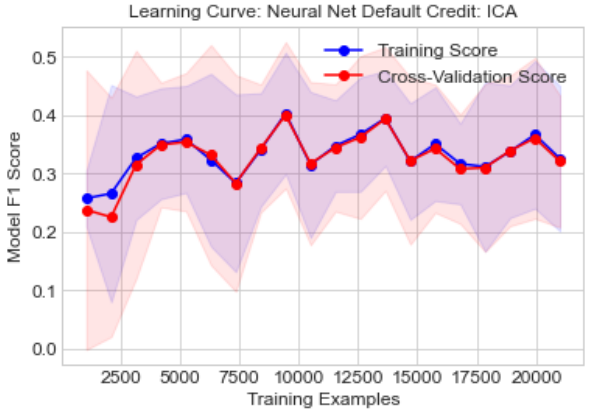
5.1 Dimensionality reduction

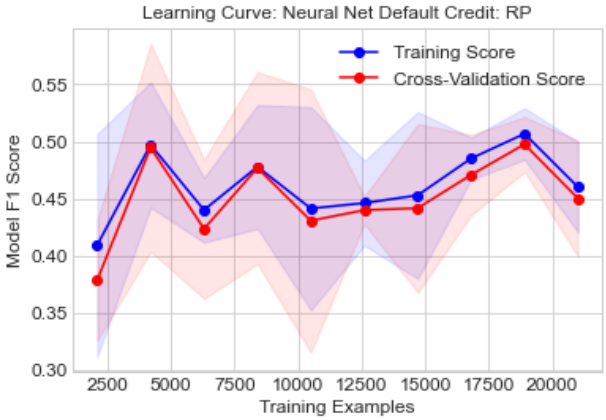
In this experiment, we apply dimensionality reduction algorithm on our default credit card dataset, and then rerun the neural network on the newly projected data. The NN has 1 layer with 50 neurons and uses sigmoid activation function. The Full data with all features is used as benchmark. Learning curves are plotted and their performances are shown in Table 4. We can see that after dimensionality reduction, F1 and accuracy scores both decreases. All 4 methods reduced accuracy with no more than 2%. However, PCA and RFC stand out by only slightly decreases the F1 score as well. The f1 score decrease for PCA is 5.4% and for RFC is 2%. ICA has low f1 score for different sample sizes. This is shown in the high AUC of PCA and RFC as well. With respect to training time, ICA and PCA are the fastest. Overall, PCA and RFC did best in this experiment.

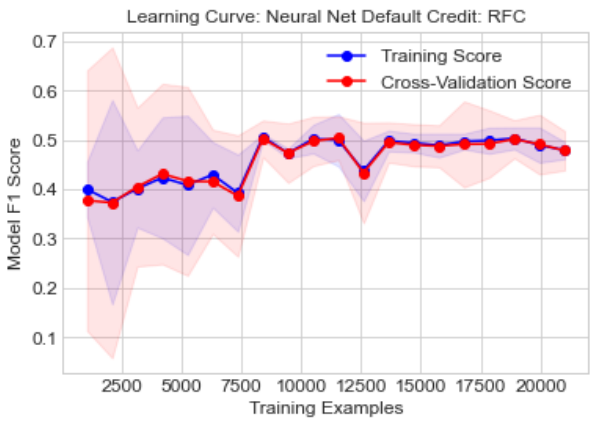
Without new features	All features	PCA	ICA	RP	RFC
training time	12.0667	5.8835	3.7505	15.9928	8.5767
prediction time	0.0071	0.0062	0.0063	0.0086	0.0066
f1 score	0.4879	0.4613	0.3225	0.4293	0.4783
accuracy	0.8213	0.8150	0.8058	0.8166	0.8127
precision	0.6737	0.6458	0.7273	0.6832	0.6534
recall	0.3824	0.3588	0.2072	0.3130	0.3773
AUC	0.6647	0.6515	0.5924	0.6360	0.6591

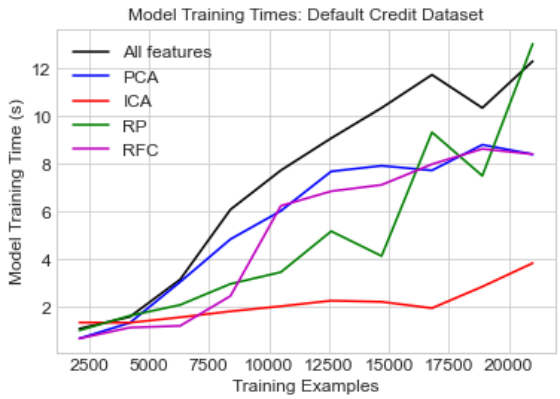
Table 4

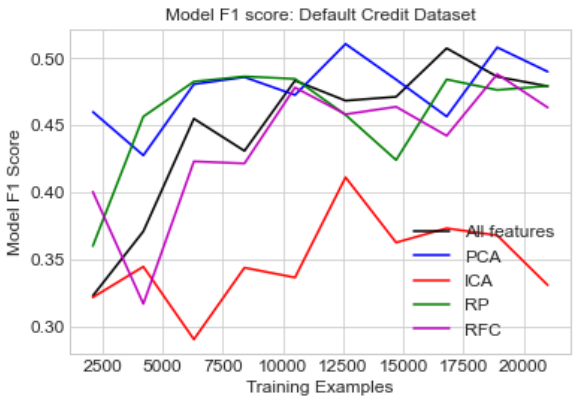










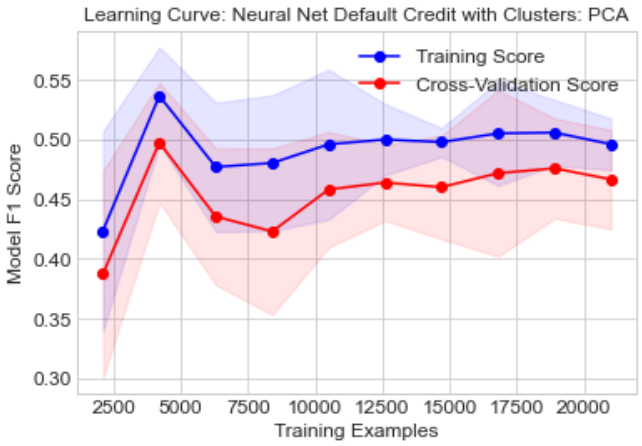


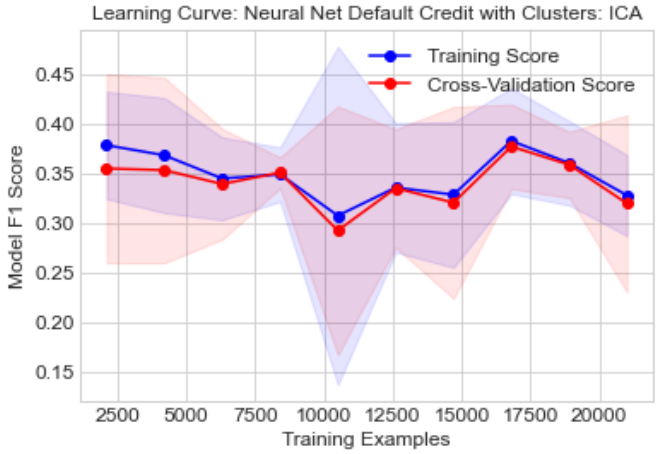
5.2 Dimensionality reduction with clustering

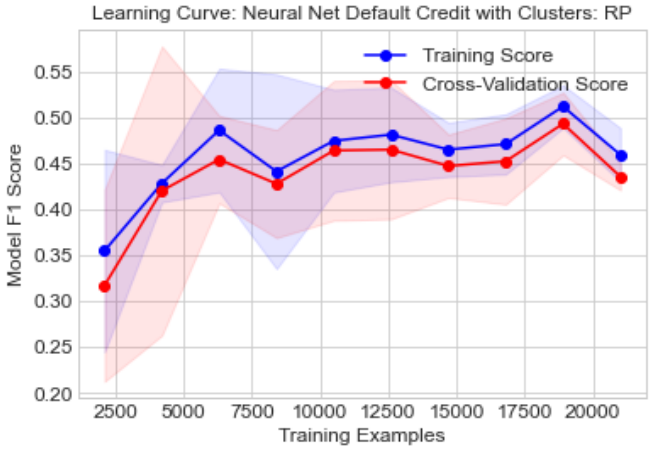
In this section, we first applied the 2 clustering algorithms on the default credit dataset. The clusters were added back to the dimension reduced datasets as new features. And then, we rerun the neural network on the newly projected data. The NN has 1 layer with 50 neurons and uses sigmoid activation function. The data with all features is used as benchmark. Learning curves and performances are shown in Table 5. With new features, F1 and AUC scores still decreases vs. benchmark. However, PCA and RP achieved slightly higher accuracy than benchmark. With respect to training time, ICA is still the fastest. Compared to 5.1, we can see that adding new features increased f1 score for all features, PCA, ICA and RP; and increased accuracy for PCA, ICA, RP, and RFC. This means the clustering algorithms extracted some information from the original dataset and helped the NN learner to increase its performance. PCA and RFC still did best in this experiment.

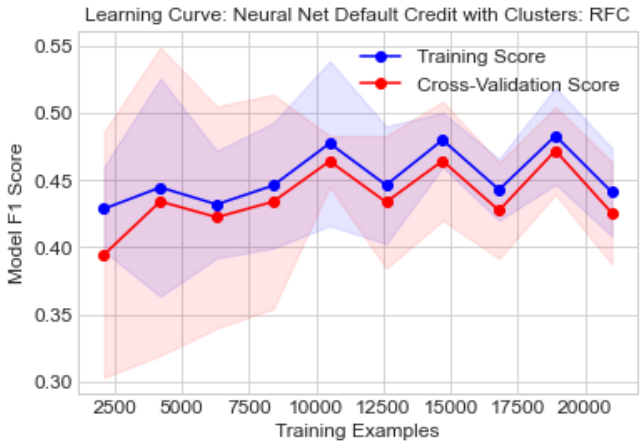
With new features	All features	PCA	ICA	RP	RFC
training time	16.6526	11.2075	8.1552	14.4304	11.4694
prediction time	0.0069	0.0101	0.0068	0.0062	0.0063
f1 score	0.4965	0.4727	0.4252	0.4601	0.4666
accuracy	0.8188	0.8200	0.8101	0.8219	0.8153
precision	0.6301	0.6772	0.6632	0.6776	0.6633
recall	0.4096	0.3630	0.3129	0.3483	0.3599
AUC	0.6713	0.6568	0.6334	0.6511	0.6535

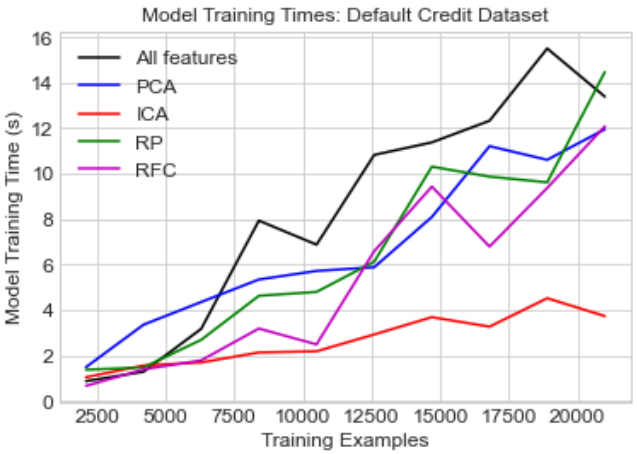
Table 5

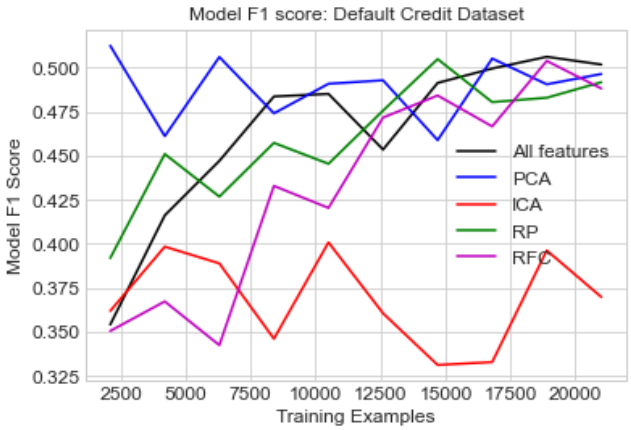












6. Reference

[1]. Phishing Websites Data Set: https://archive.ics.uci.edu/ml/datasets/phishing+websites

[2]. Default of credit card clients Data Set: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

[3]. K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks. https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

[4]. Selecting the number of clusters with silhouette analysis on KMeans clustering. https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html

[5]. V measure: an homogeneous and complete clustering. https://towardsdatascience.com/v-measure-an-homogeneous-and-complete-clustering-ab5b1823d0ad

[6]. Expectation Maximization Explained. https://towardsdatascience.com/expectation-maximization-explained-c82f5ed438e5

[7]. Probabilistic Model Selection with AIC, BIC, and MDL. https://machinelearningmastery.com/probabilistic-model-selection-measures/

[8]. Principal component analysis. https://en.wikipedia.org/wiki/Principal_component_analysis

[9]. Independent component analysis. https://en.wikipedia.org/wiki/Independent_component_analysis

[10]. Random Projection. https://en.wikipedia.org/wiki/Random_projection#:~:text=In%20mathematics%20and%20statistics%2C%20random,when%20compared%20to%20other%20methods.