UNSUPERVISED LEARNING AND DIMENSIONALITY REDUCTION

Dixon Domfeh
GTID: 903658462

1.0 Introduction

This report provides a summary of a comprehensive analysis conducted on two unsupervised clustering algorithms and four dimensionality reduction techniques. The clustering algorithms are K-Means and Expectation Maximization (EM)/Gaussian Mixture Model (GMM). We apply Principal Component Analysis (PCA), kernel Principal Component Analysis (kPCA), Independent Component Analysis (ICA) and Randomized Gaussian Projection (RGP) dimensionality reduction techniques on two interesting data sets. We further perform clustering using the reduced data and pass the output to a neural network for a final classification prediction problem. We assess the performance and fit of the above-mentioned algorithms across several metrics. The two data sets are the customer Churn and Exchange Traded Funds (ETF) datasets from Assignment 1. The Churn dataset set is particularly interesting as it's a highly imbalanced data set. The ETF is a time series format with a lot of noise. As such it can be very challenging for machine learning algorithms to make useful predictions from the data without overfitting the noise. We would like to stress that the performance of the algorithms in this report are by no means generalizable since the results from our experiments largely depends on the type of dataset been used.

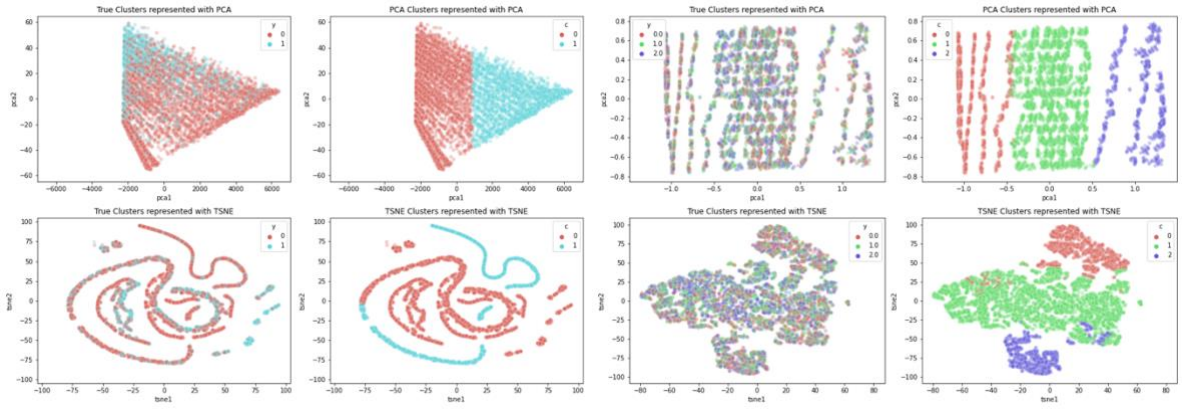
2.0 K-Means Clustering Algorithm

The k-Means approach to clustering is done by dividing data into disjoint clusters based on the mean μ_j, of samples with each cluster. The separation of data into groups with equal variance is achieved by minimizing the within-cluster sum-of-squares or inertia. K-Means uses Euclidean distances from the data points to a centroid to calculate the sum of squared deviations from the centroid and tries to minimize is as:

$$\sum_{i=0}^{n} \min_{\mu_j \in C} \left(\|x_i - \mu_j\|^2 \right)$$

In this experiment, we first fit the two datasets with the k-Means. Since the ground truths (i.e., the labels of our datasets) are known, we choose the number of components needed to fit a k-Means to the data apriori. Figure 1(a)-(b) represents the clustering results for the Churn and ETF datasets respectively.[1]Knowing the "true" labels that corresponds to the features space apriori in a semi-supervised sense clearly help k-Means separate the data into the appropriate clusters with clear separation as shown in Fig 1(a)-(b). There are three discrete labels for the ETF which represent "BUY", "SELL", or do "NOTHING". For the Churn data there are two labels – "Yes" or "No".

[1] Note that the PCA and tSNE reduction techniques are only used here to represent the data on a 2-dimensional plot. No data reduction has been performed yet.



(a) Churn data (b) ETF data

Figure 1: K-Means clustering un-reduced datasets

We try to gauge the quality of the clusters formed by K-Means by extrinsic measures that compare the ground truth labels assignment to the new clusters. We evaluate clustering performance based on homogeneity (each cluster contains only members of a single class), completeness (member of a given class are assigned to same cluster), v-measure (harmonic mean of homogeneity and completeness), Adjusted Mutual Information score (AMI), and Adjusted Rand Index (ARI) and report in Table 1 below. From Table 1, we show that the clusters performed poorly when lined up with the true labels. Perhaps, choosing the right number of clusters based on model complexity (rather than using number of true labels) coupled with dimensionality reduction can improve the quality.

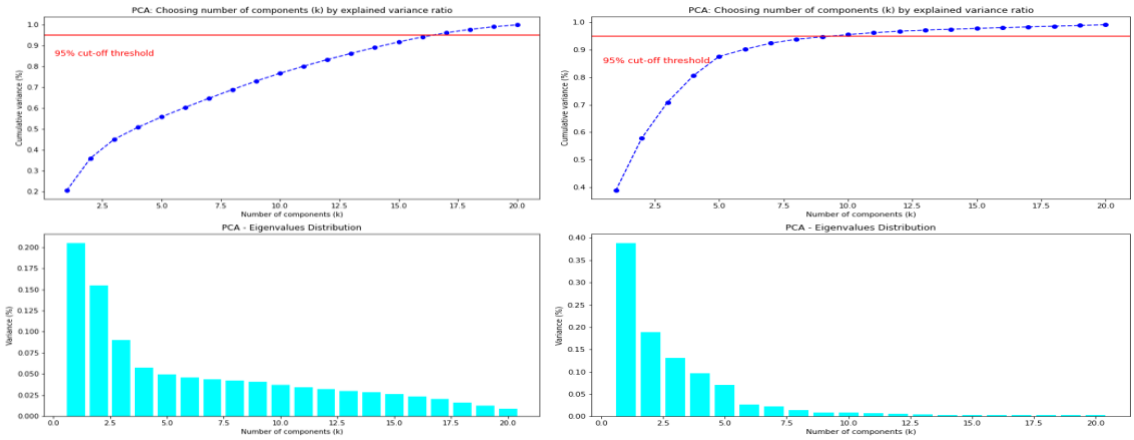
	Homo	Compl	v-measure	ARI	AMI	Silhouette	clusters
K-Means (Churn dataset)	0.004	0.004	0.004	-0.002	0.004	0.143	2
K-Means (ETF dataset)	0.016	0.016	0.016	0.018	0.015	0.160	3

Table 1: Assessing the quality of k-Means clusters against true labels

K-Means + PCA Algorithm: Next, we run the PCA algorithm to extract the components of the features space that explains most of the variance in the data and re-apply K-Means clustering. Running dimensionality reduction algorithms such as PCA is expected to increase the internal cohesion within clusters and speed up computations. To be able to realize the effect of all dimensionality reduction techniques discussed in this report, it is important to know the original number of input features in each of our dataset. There are 25 and 39 input features (excluding labels) in the Churn and ETF dataset respectively. We choose the number of principal components (k) needed to represent most efficiently the input space by expressing the amount of variance explanation we expected from the new reduced dataset. In our experiment we set the amount of explained variance to 95%. Figure 2(a)-(b) shows some plots for selecting (k). For instance, to get an explained variance of 95%, we will need 16 components (which is less than the original 25 feature space). The number of components needed is calculated as the cumulative sum of the eigen values. It is noteworthy that the eigen distribution in Figure 2(b) for the ETF dataset is relatively highly concentrated around lower number of components (x-axis). This explains why fewer principal components (only 9 in this case) are needed to explain 95% of the variance in the dataset. It also be inferred from the relative components that the Churn dataset is complex due to its imbalance nature.

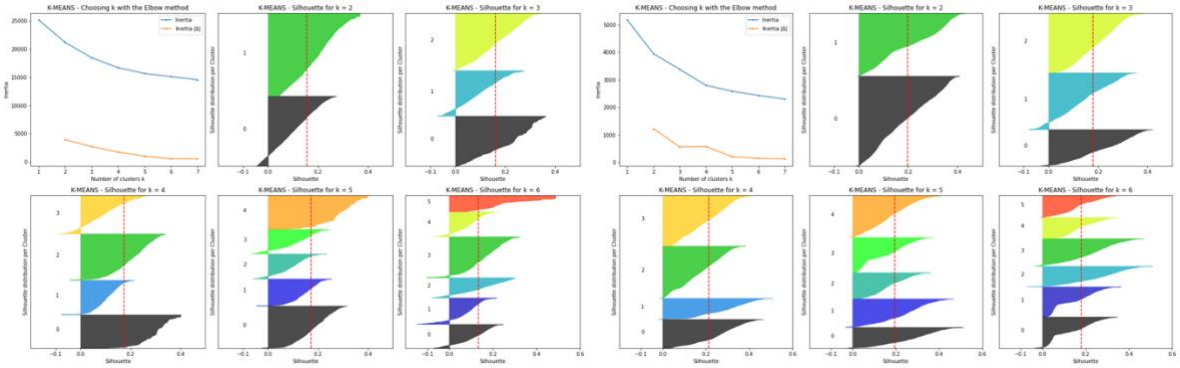
The number of clusters to be formed from the PCA-reduced dataset is determined from the model complexity plots shown in Figure 3(a)-(b). Here, we choose number of clusters based on the elbow method (using inertia) and silhouette score. Ideally, we would like a lower inertia score because it indicates tighter clusters, while we would prefer higher silhouette scores. For instance, we choose a cluster of four for PCA-

reduced churn dataset based on the afore-mentioned preference. We fit the reduced dataset to the k-Means algorithm as present the result in Figure 4(a)-(b).



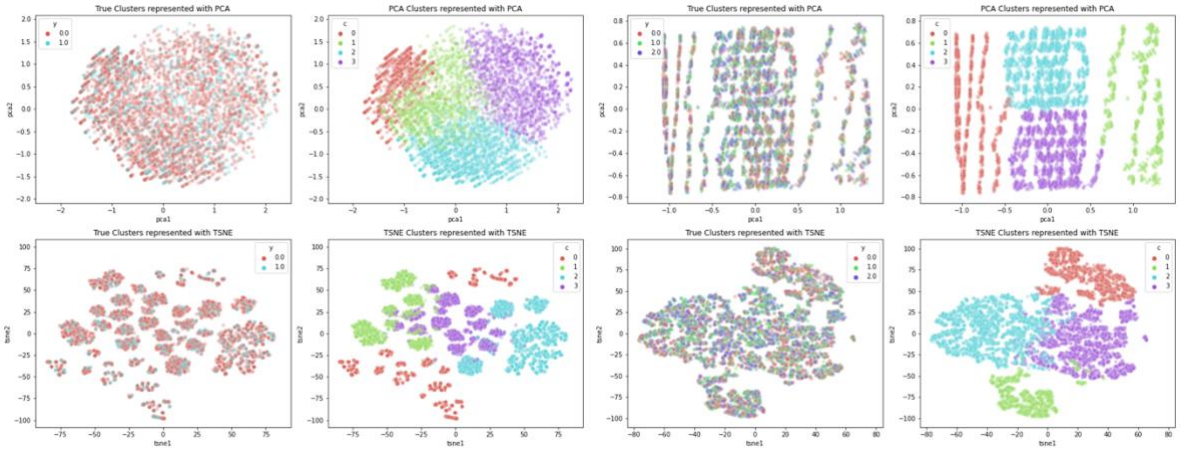
(a) Churn data (b) ETF data

Figure 2: Choosing (k) for PCA reduced datasets



(a) Churn data (b) ETF data

Figure 3: Choosing number of clusters for PCA-reduced datasets



(a) Churn data (b) ETF data

Figure 4: K-Means clustering of PCA-reduced datasets

The PCA-reduced data used for k-means clustering in Figure 4 show four clearly distinct clusters which may seem a departure from the true labels observed in the data. However, the four clusters reveal some interesting subgrouping within the larger binary labels (i.e., Yes and No in the case of the churn dataset). It can be intuited that the 4 k-means clusters can be linearly separated into two larger groups. In fact, we show that the 4 clusters improve in the performance as reported in Table 2 below. For instance, the performance in terms of the v-measure and silhouette score much improvement in the case of PCA-reduced data with 4 clusters relative to the baseline model with 2 clusters (see Table 2.1-2.2 below)[2].

Churn dataset	Homo	Compl	v-measure	ARI	AMI	Silhouette	clusters
K-Means (Baseline)	**0.004**	**0.004**	**0.004**	**-0.002**	**0.004**	**0.143**	**2**
K-Means (ICA)	0.078	0.029	0.043	0.015	0.041	0.090	5
K-Means (PCA)	**0.169**	**0.071**	**0.100**	**0.047**	**0.099**	**0.172**	**4**
K-Means (kPCA)	0.177	0.075	0.105	0.057	0.104	0.177	4
K-Means (RGP)	0.093	0.082	0.087	0.008	0.087	0.220	2

Table 2.1: Assessing the quality of reduced k-Means clusters against true labels (Churn dataset)

ETF dataset	Homo	Compl	v-measure	ARI	AMI	Silhouette	Clusters
K-Means (Base)	0.016	0.016	0.016	0.018	0.015	0.160	3
K-Means (ICA)	0.030	0.034	0.031	0.034	0.030	0.128	3
K-Means (PCA)	0.023	0.018	0.020	0.025	0.019	0.183	4
K-Means (kPCA)	0.024	0.019	0.021	0.027	0.019	0.189	4
K-Means (RGP)	0.012	0.012	0.012	0.013	0.010	0.185	3

Table 2.2: Assessing the quality of reduced k-Means clusters against true labels (ETF dataset)

K-Means + Kernel PCA Algorithm: The kernel PCA algorithm is a non-linear dimensionality reduction version of PCA. We tried four different kernels all four kernels available – cosine, gaussian (rbf), sigmoid, polynomial and chose the kernel which gives a 95% explained variance with the minimum number of components. The final kernel of choice is the sigmoid kernel. The model complexity plot for kernel PCA is reported in Appendix. Kernel PCA reduced clusters also show 4 clusters which look like the ones presented in Figure 4 above for both datasets.[3] Notably, there is an increase in performance for the kPCA over the linear PCA as reported in Tables 2.1-2.2. Due to its non-linear kernel, kPCA can map out the nonlinearity in the data much better!

K-Means + RGP Algorithm: The randomized Gaussian projection was used in this report. RGP is ideal for very high dimensional dataset. The eigen vectors are chosen at random making it very efficient and cheap. This is in opposite comparison to PCA where eigen vectors computation is expensive. Due to the limited features used in our experiment, the minimum dimension for RGP which is chosen by the Johnson-Lindenstrauss lemma was not possible. We tried a brute force hyperparameter tuning by running several combinations of eps and n_component of the RGP algorithm in sklearn and measure the absolute difference in distance between the transformed features and the actual. Figure 5 shows a plot of two of such model complexities for the ETF dataset.

The reduced data from RGP coupled with k-means reveal 2 clusters for churn data set and 3 clusters for the ETF data set which is consistent with the numbers of labels in the data (see Figure 6 below). However, its ability to map the data is poor compared to the other dimensionality reduction algorithms discussed here (refer to Table2.1-2.2). We run the RGP algorithm many times to confirm the notion that the projection matrix follows a Gaussian distribution (see Figure 6 below). This shows that the random project does not have a true orthogonal matrix but close to being the true orthogonal projection like in PCA.

[2] All scores are evaluated on a test data set.
[3] Plot looks similar. We do not report it since we want to keep the report brief.

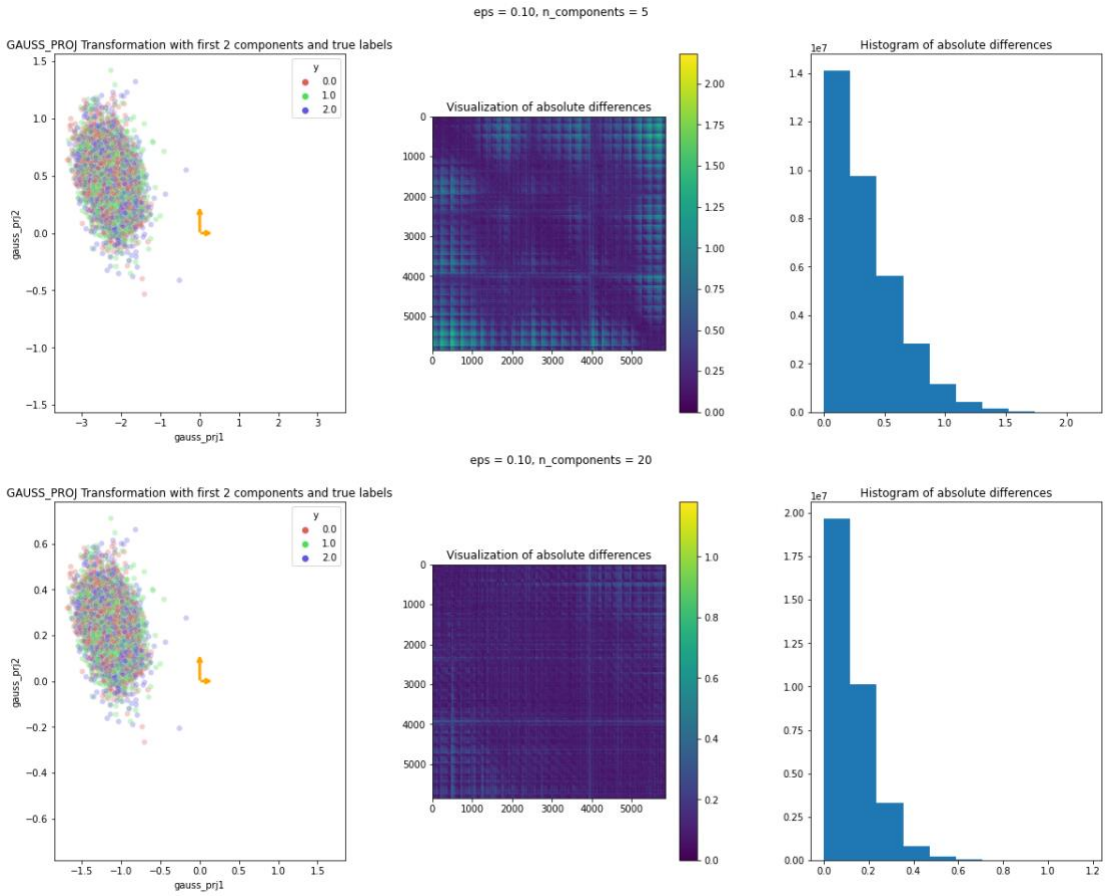


Figure 5: Choosing number of components based on absolute distance difference.

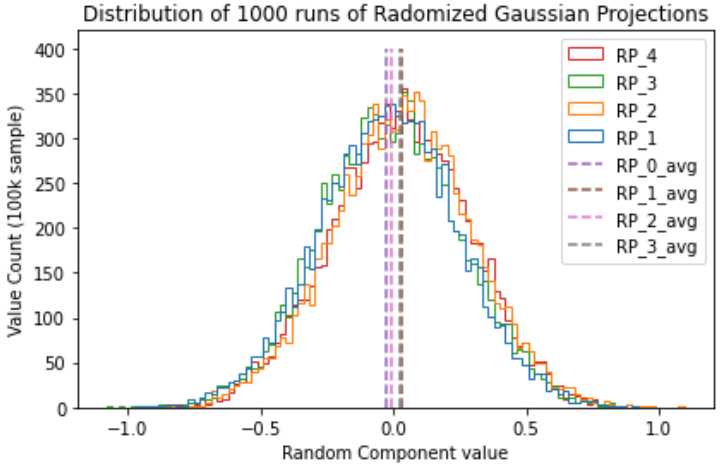
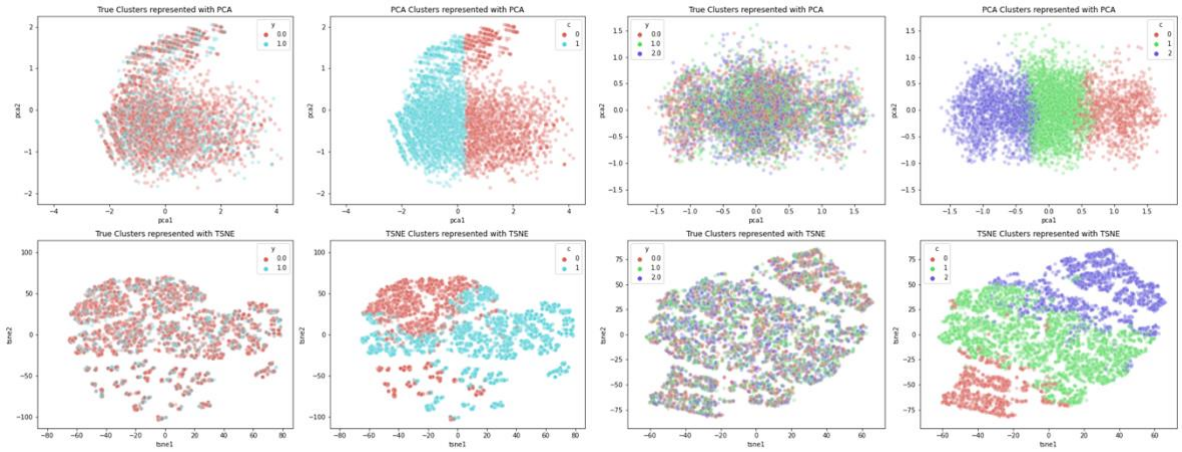


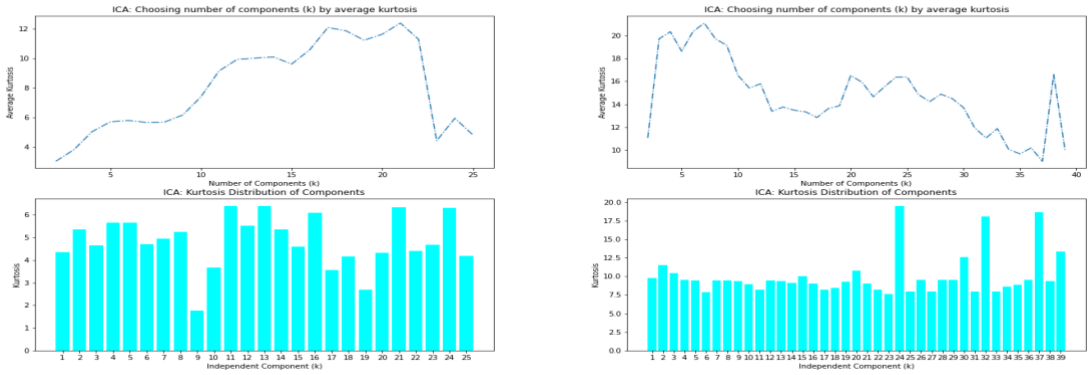
Figure 6: Distribution of transformation matrix of RGP for ETF dataset



(a) Churn data (b) ETF data

Figure 7: K-Means clustering of RGP-reduced datasets

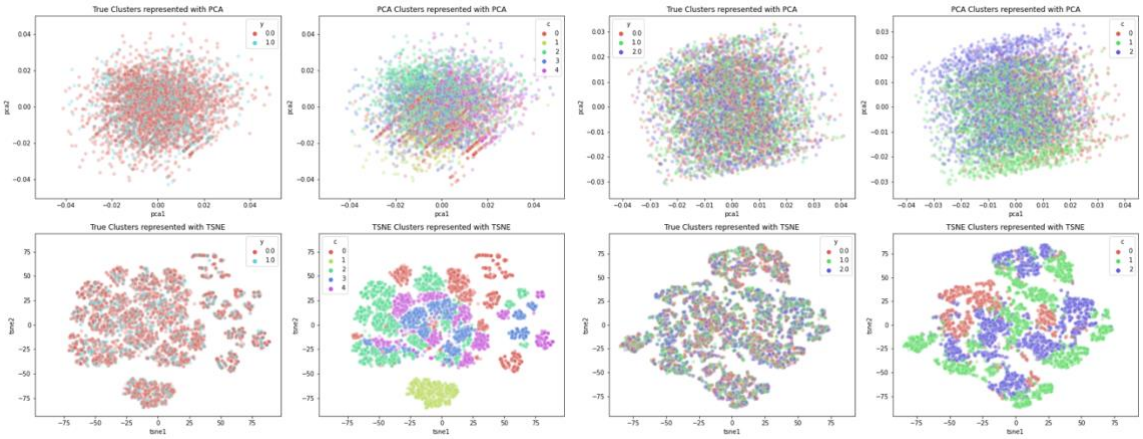
K-Means + ICA Algorithm: The ICA algorithm attempts to find statistically independent components via a linear combination of input features. It is built on the idea of blind source separation used to recover the original independent signals from a recording of mixed sounds. We determine the number of components needed to reduce our datasets by assessing the average kurtosis of the components. Figure 8 below present the plots. The optimal number of components is selected based on the number of components that provides highest average kurtosis. In implementation, we use the Fast ICA. The highest kurtosis criterion is to find the best direction in the feature space which corresponds to projections resulting in high non-Gaussianity. For the churn dataset higher components are needed relative to the ETF dataset, highlighting the same complexity observed in the PCA reduction.



(a) Churn data (b) ETF data

Figure 8: Choosing number of components for ICA-reduced datasets

The ICA-reduced k-means clusters shown in Figure 9 below suggest 3 clusters which is consistent with the true label classed of 3. However, that interpretation is not meaningful. A closer look at the performance report in Tables 2.1-2.3 show that ICA's performance is lower than that of PCA when its clusters are compared to the ground truth.



(a) Churn data (b) ETF data

Figure 9: K-Means clustering of RGP-reduced datasets

3.0 EM Clustering Algorithm

EM algorithms take a "soft" approach to clustering. It groups data into clusters probabilistically. We show the model selection and complexity of fitting the EM algorithm to the two datasets in Figure 10 below using the Bayesian Information Criterion (BIC). [4]

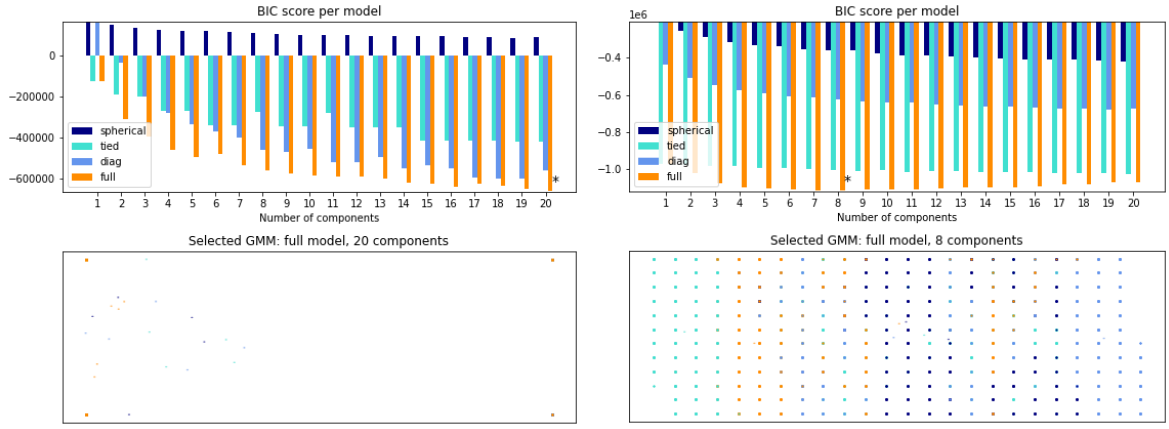
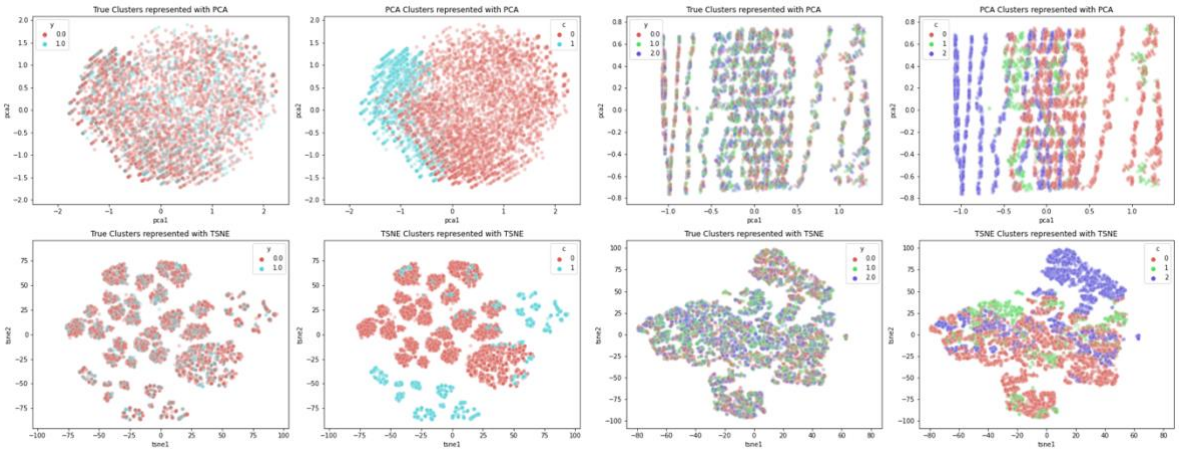


Figure 10: Choosing number of components for EM on datasets

The results of fitting the above selected models in Figure 9 to the un-reduced dataset is reported in Figure 11 below.

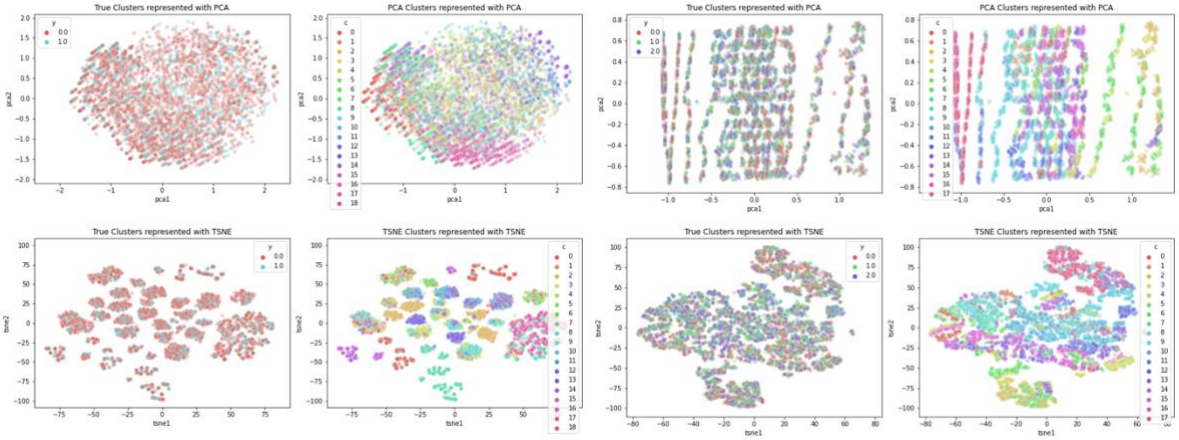
[4] * in Figure 7 show the optimal BIC score.



(a) Churn data (b) ETF data

Figure 11: EM clustering un-reduced datasets

EM + PCA Algorithm: The clusters formed from the PCA-reduced data show more granular subgroupings in the dataset which may not be relevant for predicting the labels. For instance, the EM show 18 clusters for the Churn dataset (see Figure 12 below). To get a sense of the quality of these clusters, we compare them to be ground truth by extrinsic metrics presented in Table 3.1-3.2
below.



(a) Churn data (b) ETF data

Figure 12: EM clustering of clustering of PCA-reduced datasets

Churn dataset	Homo	Compl	v-measure	ARI	AMI	Silhouette	components
GMM (Base)	**0.004**	**0.004**	**0.004**	**-0.018**	**0.003**	**0.103**	**2**
GMM (ICA)	0.153	0.038	0.061	0.039	0.057	-0.009	15
GMM (PCA)	**0.071**	**0.016**	**0.025**	**0.000**	**0.022**	**0.028**	**19**
GMM (kPCA)	0.150	0.032	0.053	0.020	0.049	0.017	18
GMM (RGP)	0.187	0.041	0.067	0.014	0.063	0.012	20

Table 3.1: Assessing the quality of reduced EM clusters against true labels (Churn dataset)

In general, the performance of the EM clusters both with dimensionality reduction methods and un-reduced data is relatively lower as compared to the k-means on the data set used. Based on the performance review in Table 3.1 we decide not to present more on the findings on the EM algorithm's results since there is nothing strikingly different. Finally, we assess the quality of the dimensionality reduction algorithms based

on how well they can reconstruct the feature space after reducing the data by measuring the mean squared error (MSE). Table 3.3 reports our results. From the table below, PCA shows the least error. This can confirm why PCA shows the best performance relative to the other algorithms when paired with k-Means and EM on the ETF dataset. The ICA algorithm performs best on the churn dataset. While RGP performs the least on both datasets. This is due to the low dimensionality of the feature space of our dataset.

	ETF		**Churn**	
	No. of Components	**Reconstruction Error (MSE)**	**No. of Components**	**Reconstruction Error (MSE)**
PCA	7	0.00183	21	3.746e-06
Kernel PCA	9	0.00128	16	0.01130
ICA	9	0.0138	19	0.07887
GRP	15	0.1487	14	0.12703

Table 3.3: Reconstruction error ETF datasets

4.0 Neural Networks

The results of performing classification predictions with the reduced data alone and adding them as additional feature are presented in the following tables using a neural network. We evaluate the performance using the churn dataset. As presented in Table 4.1, the performance using only the reduced data show no added improvement in prediction. In some cases, the performance in even reduced in terms of ROC AUC score. For instance, the F1-score on the minority class ("No") is practically zero for KPCA and ICA. When we add the predicted clusters from the reduction algorithms as additional feature, there is only a marginal improvement in performance.

Feature Reduction Methods					
Metric	**Feature Only**	**PCA**	**Kernel PCA**	**ICA**	**RGP**
Yes (F1-score)	0.86	0.85	0.85	0.85	0.85
No (F1-score)	0.57	0.56	0.00	0.00	0.50
ROC AUC	0.8382	0.836	0.592	0.499	0.818
Training time	1.638 sec	1.5738 sec	1.618 sec	1.648 sec	1.589 sec
Query time	0.0014 sec	0.0025 sec	0.0014 sec	0.0020 sec	0.0012 sec

Table 4.1: Neural network classification results for reduced churn dataset.

Metric	**Feature Only**	**Features + (K-means clusters from ICA)**	**Features + (K-means clusters from PCA)**	**Features + (K-means clusters from kPCA)**	**Features + (K-means clusters from RGP)**
Yes (F1-score)	0.8600	0.86	0.86	0.86	0.86
No (F1-score)	0.5700	0.57	0.57	0.57	0.59
ROC AUC	0.8382	0.836	0.837	0.836	0.836
Training time	1.638 sec	1.667 sec	1.859 sec	1.795 sec	1.67 sec
Query time	0.0014 sec	0.0016 sec	0.0018 sec	0.0017 sec	0.0017 sec

Table 4.2: Neural network classification results for churn dataset with addition features from k-means clusters.

It is noteworthy to point that RGP surprisingly, shows an improvement in the F1-score for minority class (refer to Table 4.2 below) for the k-Means clusters. We performed a similar analysis using the EM clusters as additional inputs the neural net. In terms of training and query time, there is no distinct difference across all reduced datasets. The results are not quite different (see Table 4.3).

Metric	Feature Only	Features + (EM clusters from ICA)	Features + (EM clusters from PCA)	Features + (EM clusters from kPCA)	Features + (EM clusters from RGP)
Yes (F1-score)	0.86	0.86	0.85	0.85	0.86
No (F1-score)	0.57	0.57	0.55	0.56	0.57
ROC AUC	0.8382	0.836	0.836	0.836	0.835
Wall time (training)	1.638 sec	1.718 sec	1.661 sec	1.665 sec	1.661 sec
Wall time (query)	0.0014 sec	0.0019 sec	0.0016 sec	0.0015 sec	0.0015 sec

Table 4.2: Neural network classification results for churn dataset with addition features from EM clusters.

5.0 Conclusion

Thus far, this report has presented our findings of applying four dimensionality reduction algorithms to two data set and assessed their usefulness in solving a classification and clustering problems. The results based on our observation reveal that the reduced input features alone are not sufficient to improve prediction on a classification problem. When the clustered features are added as additional features, the marginally improve prediction. Our conclusions are drawn based solely on the dataset used and therefore cannot be generalized. A suggestion for future analysis is to try out the same process outlined here on a sufficiently large data set to truly assess the power of algorithms such as randomized gaussian projections. Our dataset is specifically difficult to cluster and predict due to its imbalance nature.