Introduction of Data Used in Analysis

To examine the effects of clustering and dimensionality reduction, various experiments were performed in this analysis using the Bank Marketing Data Set and the Letter Recognition Data Set. The Bank dataset is for a binary classification task, whereas the Letter dataset is for a 26-class classification task (one for each letter of the alphabet). A significant difference between the two datasets is the class label distribution. The Bank dataset is composed of roughly 88% negative class instances, whereas the Letter dataset has near uniformly distributed classes. As such, the effects of the clustering and dimensionality reduction algorithms vary significantly between the two datasets.

Clustering Experiments

K-Means Clustering

The first clustering algorithm examined was the K-Means algorithm. This algorithm involves using mean centered clusters with some quantity ‘k’ (clusters) that is passed to the algorithm. The unsupervised nature of clustering does not lend an easy answer to what an appropriate value of k should be, and a variety of methods were used in this analysis.

The Elbow Method

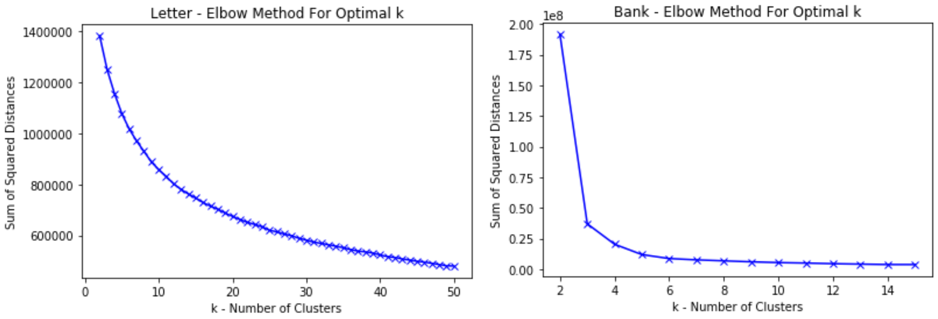


Figure 1 – Elbow Method for Distortion

As the name suggests, the elbow method involves looking for an “elbow” on a plot with a measurement of how well the clusters represent the data on one axis, and the number of clusters (k) on the other axis, as shown in *Figure 1*. The idea is to find a value of k such that the improvements in distortion become marginal (a plateau is found). For the K-Means algorithm, the sum of squared distances (i.e., distortion) is a way to measure how well the clusters are defined. For the Letter dataset, the plot was smooth, and no clear elbow was apparent, however, the Bank plot had a clear elbow at a k of **3 or 4**, which provides a value that represents k well without overfitting. As k increases towards the number of data points, distortion goes to zero, but the clustering would not be indicative of any meaningful information at that point, so an ‘elbow’ suffices.

Silhouette Plots

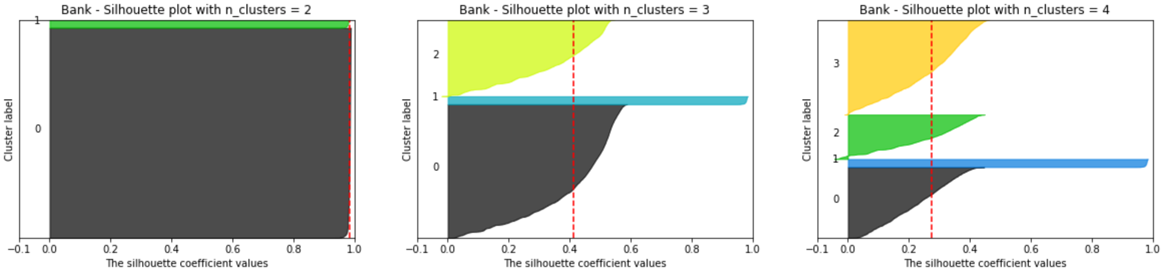


Figure 2 - Silhouette Plots (Bank)

To find a more conclusive answer for the best value for k, silhouette plots can be used to test different values. A silhouette in the context of clustering is a measure that graphically summarizes, how well clusters are formed, by calculating distances to other clusters and possible misclustering. The plots took an extensive amount of time to generate, however, they can be helpful for picking a value for k. In *Figure 2*, three silhouette plots are shown for the Bank dataset for illustration. When k is 2, we find silhouettes that very well represent the data, which makes sense because the data is meant to be used for binary classification. However, this does not necessarily indicate this is the best value for k, because one may want to find different patterns in the data than a binary separator. The elbow method and the T-SNE plots (shown later) will help show why 3 or 4 might be a better k than 2, despite the high silhouette scores for k of 2 for the Bank dataset.

In order to summarize the results of each k for silhouettes without showing each graph, I plotted the average silhouette scores for a given k for both datasets, as shown in *Figure 3*. As expected for the Bank dataset, the best value for k seems to be between **2 and 4,** while the Letter dataset remains inconclusive. It seems that K-Means has a much more difficult time clustering the Letter data. Using domain knowledge, we know that there are 26 class labels, however, the clustering does not necessarily approximate the classes alone. The value for k I selected for the Letter dataset was **30**, because, while distortion is decreasing fast at 30, the silhouette scores start to decline to unacceptable levels beyond that point. In summary, the Letter dataset appears to be a particularly hard problem for K-Means.

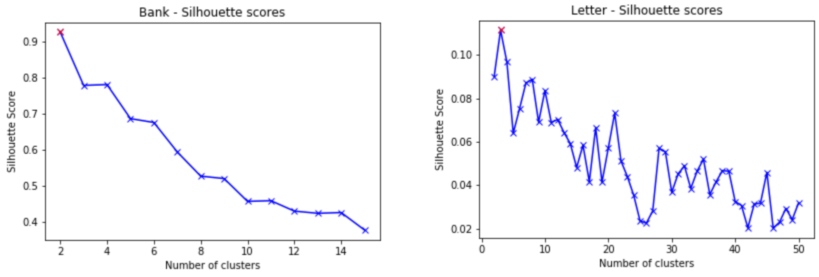


Figure 3 - Average Silhouette Summary

Expectation Maximization (EM)

EM is another approach at solving the clustering problem. The Gaussian Mixture model (GMM) from the sklearn library was used to analyze EM for the Bank and Letter datasets. Instead of picking the best value of k, the best number of components must be found, which functionally, is the same sort of problem as picking the best k. EM is focused around the probability a given instance belongs to a cluster (i.e., a "soft" boundary), rather than focusing on a centroid relationship with the nearest cluster. As such, EM can capture clusters that K-means may not be able to capture very well. Naturally, due to the difference in the way the algorithm is performed are some nuances that make finding the correct number of components different than finding the best value for k.

The Elbow Method (Again)

The elbow method can also be used for EM; however, the distortion metric is not a metric that makes sense in the context of EM. Instead, however, log likelihood or Bayes (or Akaike) information criterion (BIC/AIC) can be used to measure the probability that the number of components fits the data well. As shown in *Figure 4*, log likelihood approximates a curve that is an inverse of the BIC/AIC curve, so only one of these curves is necessary to locate an elbow. The goal is to maximize likelihood or minimize BIC/AIC, without overfitting. For the Letter dataset, there is a soft elbow starting around 20 components, and ending around 30 components, so **30** components were selected for the GMM of the Letter dataset. For the Bank dataset, a clear elbow is found again at **4** components. Silhouettes can technically be used for EM; however, it did not tend to do well with non-spherical clusters, which EM can approximate.

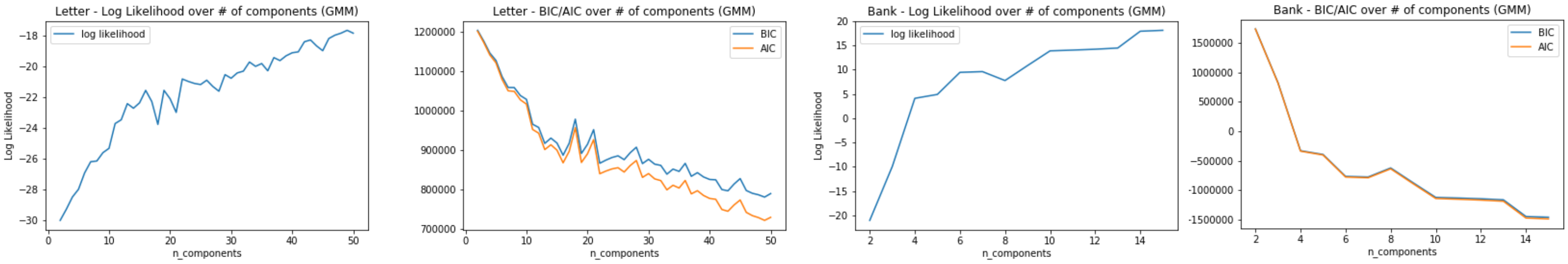


Figure 4 - Elbow Method for Likelihood

Results Visualized (T-SNE)

Now that the number of clusters to approximate for K-Means and EM have been selected, the question remains: how does one know if these are good values for clustering? One way of visualizing the clustering performance is to project the data into 2D space using a t-distributed Stochastic Neighbor Embedding (T-SNE) plot. Because the Letter dataset has 15 features, and the Bank dataset as 18 features, the T-SNE plot can do an adequate job at visualizing the data in 2 dimensions. In *Figure 5,* the left most images are colored by their true labels. For example, the ground truth visualization for the bank dataset contains yellow points for positive class instances, and purple for negative class instances. Both K-Means (with k of 3) and EM (4 components) recover the labelling well, however, they identify a deeper pattern in the dataset. There are clusters with a high proportion of positive instances, some with a medium proportion, and some with very little positive instances, and this is the pattern that the clustering algorithms found in the dataset, rather than strictly positive or negative classed areas. As expected, the Letter dataset is highly overlapping toward the middle of the plot, making the clustering challenging to perform cleanly. Nevertheless, with a k of 30 and 30 components, both algorithms do a decent job at recovering the labels (EM doing a slightly better job at avoiding bleed-in from other clusters on the outer edge). Both algorithms do not clearly stop improving after 26 clusters, even though there are 26 class labels, suggesting that neither algorithm was able to recover the classes cleanly, due to the proximity of each cluster to other clusters, especially towards the middle.

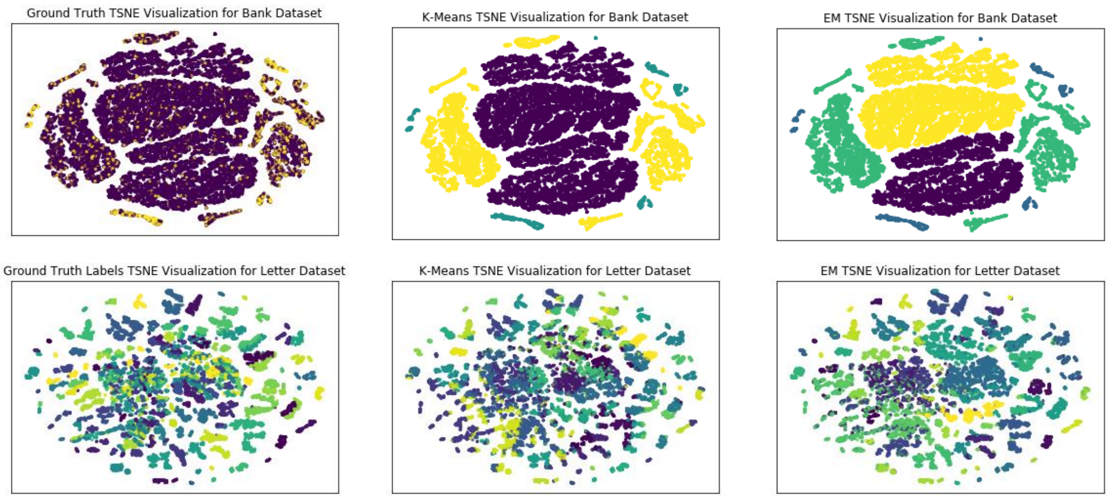


Figure 5 - T-SNE Plots for K-Means and EM

Dimensionality Reduction

Various methods can be used in order to reduce the number of features in a dataset. This is useful for removing noise from data that would otherwise affect algorithms further down the data pipeline. It is also useful to improve the processing time required, as well as the memory required to store data. Experiments were performed on four dimensionality reduction techniques for the Bank and Letter datasets.

Principal Component Analysis (PCA)

PCA is a popular method for dimensionality reduction. It involves calculating eigenvectors for each feature of the data and ranking them by order of eigenvalue, however, the number of principal components (i.e., features) to keep in the reduced dataset is a hyper-parameter that requires tuning and careful evaluation. In order to find an appropriate number of components for reducing the Letter and Bank datasets, the elbow method was performed on reconstruction error and Cumulative explained variance, which are inversely proportional, (this is similar to log likelihood and AIC/BIC).

In *Figure 6*, the Bank dataset plots show that the dataset is nearly completely reconstructed at only 2 components, and only very slightly benefits from 2 additional components, for a total number of **4** components. This implies that there are only a few principal components that matter to PCA in the Bank dataset. In contrast, each feature of the letter dataset seems to play a more important role, with a smooth curve containing no clear elbow. At roughly 12 components, however, most of the data can be reconstructed, so **12** components were used to reduce the features slightly.

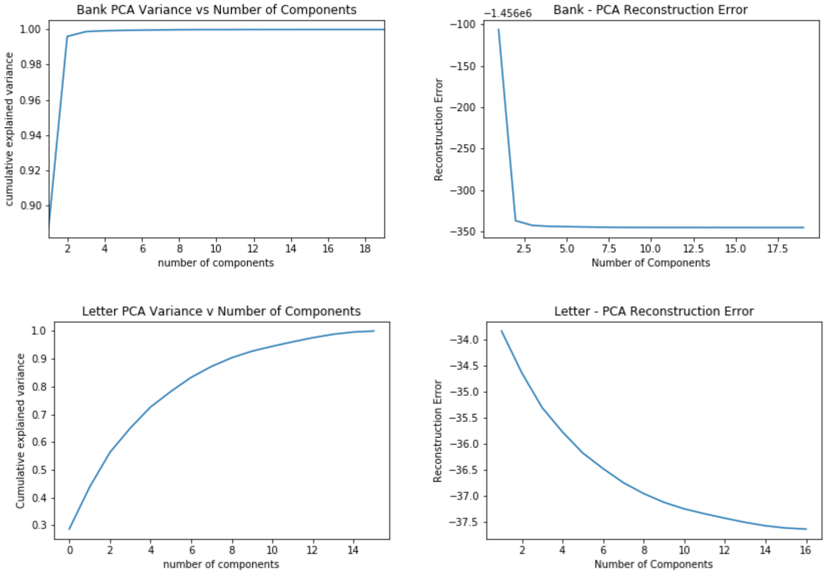


Figure 6 - Elbow Method for PCA # Components

From the plots, it's clear that the eigenvalues are sorted such that the features producing the highest eigenvalues are prioritized when the number of components is limited to less than the total number of features. PCA assumes that the eigenvectors with the highest eigenvalues are the most important features of the dataset.

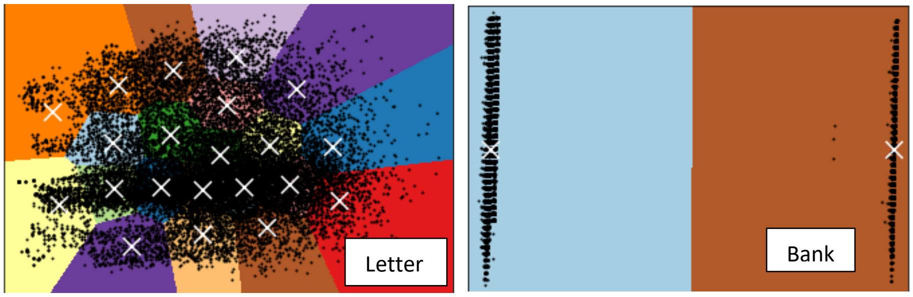


Figure 7 - Voronoi Diagram (Reduced to 2 Dim)

To visually analyze how the dimensionality reduction affects the clustering, a Voronoi diagram (shown in *Figure 7*) was plotted for each dataset using the 2 most principal components of each. Within the diagrams, even reduced to two dimensions, the clusters for the Letter dataset remain difficult to distinguish, with very little separation. Conversely, the Bank dataset has a very large amount of separation when only 2 principal components were used. This further supports the reconstruction plots in *Figure 6* and helps verify that the number of components required for the bank dataset is far smaller than what is required for the letter dataset. A more granular look at the label overlap when the Letter dataset is projected to 2 dimensions shown in *Figure 8*, where the points are colored instead of the background of the clusters, to further illustrate the difficulty of recovering the ground truth labels.

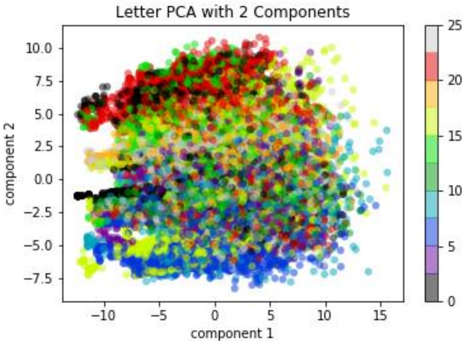


Figure 8 - Ground Truth Labels (A 'Beautiful Hairball')

Now that the datasets have been reduced, again, the clustering algorithms are tuned for each dataset. In *Figure 8*, when searching for the best value for k in K-Means, another clear elbow appears at a value of 3 clusters, where 4 may also be appropriate. This is consistent with the silhouette score as well, as the average silhouette coefficient for 2-4 is very high. The final value of **3** was selected using the T-SNE as a tie breaker.

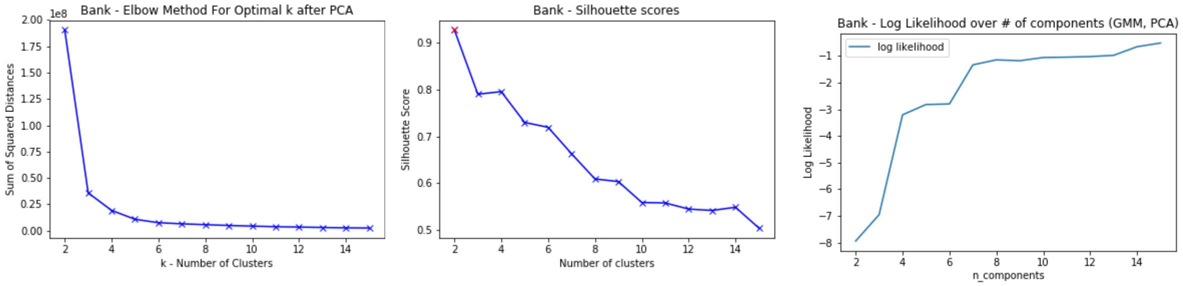


Figure 9 – Finding best # of components and k for the Bank Dataset (PCA)

For the number of components to use for EM, two elbows show up in the graph, making the decision more difficult. Again, using domain knowledge of the graph and T-SNE (shown later), **4** was the most reasonable number of components to use for EM on the bank dataset.

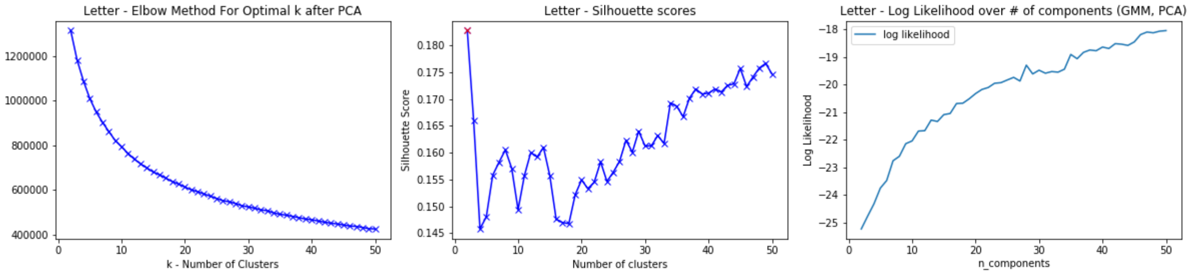


Figure 10 – Finding best # of components and k for the Letter Dataset

Once again, the Letter dataset proved more difficult in finding the right value for k and for the number of components. In, general, even with a slightly reduced dataset, the 'best' value for k remains elusive. The dimensionality cannot be reduced significantly with PCA because a significant amount of data loss occurs, so the results are not far from the original clustering results. A value of **33** was chosen for k based on the jump in silhouette scores, and the slowdown of the descent of the distortion at k of 33 in *Figure 10*. Similarly, a slight elbow at 28 was found in the rightmost plot of *Figure 10*, so the number of components used for EM clustering was **28**.

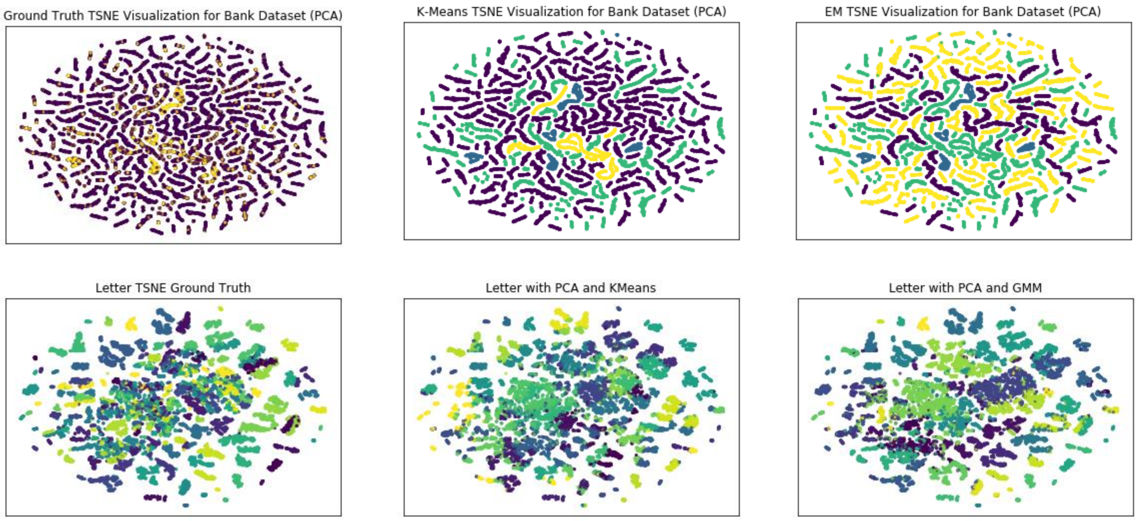


Figure 11 – T-SNE Plots for K-Means and EM with PCA Reduction

After using T-SNE to plot the clustering results in 2D after performing PCA reduction, the Bank dataset showed a drastic transformation in shape, but not necessarily clustering results. The clusters were like the results in *Figure 5,* regardless of how the points projected into the plane. The clustering was nearly able to retrieve the same results as the ground truth labelling, however, with more imprecision (all or nothing clusters, rather than mixed). The Letter data clustering remained largely unchanged from the results of *Figure 5*, which is unsurprising considering the data was not reduced by many dimensions.

Independent Component Analysis (ICA)

ICA is another tool that can be used in order to determine how to reduce the dimensionality of a dataset. Instead of maximizing eigenvalues and finding eigenvectors, ICA instead assumes each feature is non-gaussian, and that they are independent from another. In layman's terms, ICA is viewing each feature as its own independent characteristic that is evaluated by its own merits. A common analogy used to describe ICA is how humans interact with the world (e.g., the "cocktail party problem"). In a very noisy environment, humans can hold a conversation by focusing on a single signal, i.e., the voice of the person they are talking to, even though there is many, sometimes louder, signals around them.

ICA on its own, however, does not provide the best combinations of features, it is only a tool for evaluating how non-gaussian a feature or collection of features is. To determine the right number of components for a given dataset, experiments must be performed.

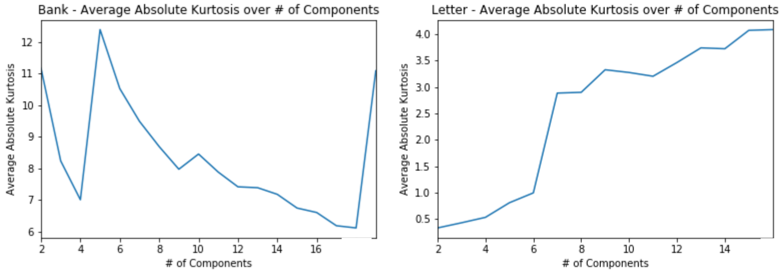


Figure 12 – Average Absolute Kurtosis over # of Components for ICA

To evaluate the number of components for ICA, kurtosis (a measure of how “non-gaussian” a distribution is), specifically mean *absolute* (kurtosis can be negative or positive, but both are treated the same) kurtosis, can be plotted as the number of components are increased. For the Bank dataset, there is a peak of kurtosis found at 5 components, and as such, **5** components were used for the FastICA implementation in sklearn on the Bank dataset. The Letter dataset, on the other hand, has a maximum average kurtosis with *all* the features included. The purpose of this experiment is to reduce dimensions, and if all the features were included, then no reduction would be performed. In lieu of this, a component number of **9** was chosen because most of the kurtosis is achieved at 9, while still performing substantial dimensionality reduction, and the effects were able to be observed on the clustering and neural network (shown later) experiment results.

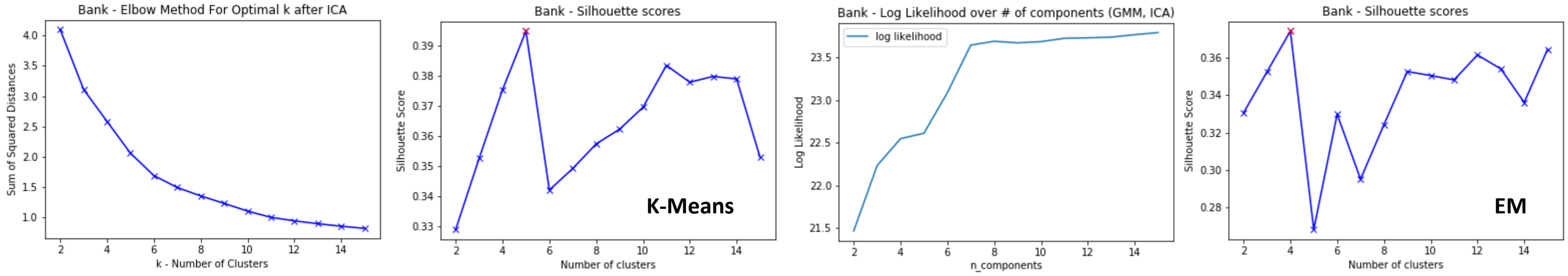


Figure 13 - Finding best # of components and k for the Bank Dataset (ICA)

The results from *Figure 13* are different than the tests from the full Bank dataset and the PCA reduced dataset. The elbow method was insufficient to find a clear value for k; however, a slight elbow appears at a k of 5. The silhouette method, however, provides clear best results at a k of **5** with respect to silhouettes. For EM, two elbows are found again, however, the silhouette scores suggest that the first elbow of **4** components is the marginally better choice.

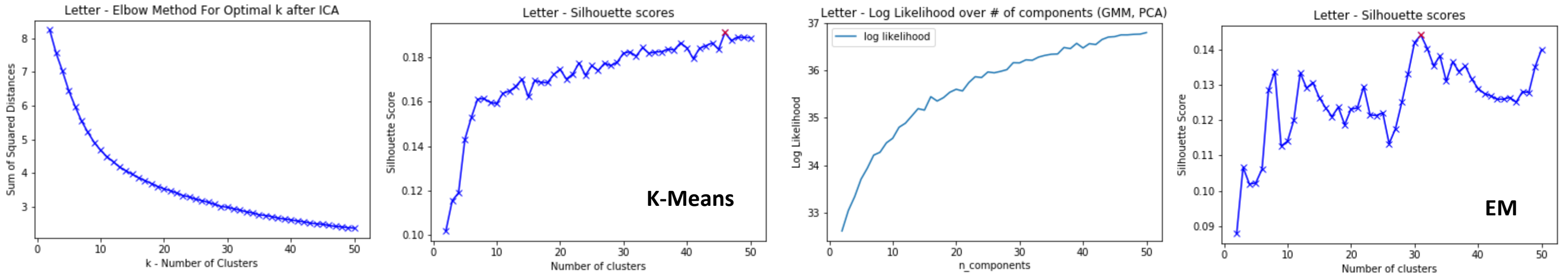


Figure 14 - Finding best # of components and k for the Letter Dataset (ICA)

The Letter dataset has slightly different results from the previous clustering experiments. In *Figure 14*, the best k is very difficult to determine. From the data, it appears that increasing clusters to 45 would provide good results, however, with the domain knowledge of there only being 26 letters, and by experimenting with T-SNE plots. 45 is likely too many clusters and may result in overfitting the data if the goal is to retrieve the original class labels (which it is not, necessarily). Nevertheless, a k of **45** was selected considering the results. For EM, a more reasonable best silhouette score was achieved at **31** components, and no clear elbow appeared in the likelihood plot.

Again, the clustering was visualized using T-SNE plots on the ICA transformed data. Results for the Bank dataset are like the results for PCA in *Figure 11*, however, the positive labels seem to be more dispersed than they were in the PCA ground truth graph. The clustering seems to again coarsely retrieve the original labels. For the Letter dataset, a k of 45 surprisingly captures the nuance of the original ground truth labels. I hypothesize that its clustering different forms of letters as two distinct clusters, which helps it capture the essence of the labels. EM seems to be clustering something different than the class labels alone and is more coarsely clustering the data in the middle of the plot.

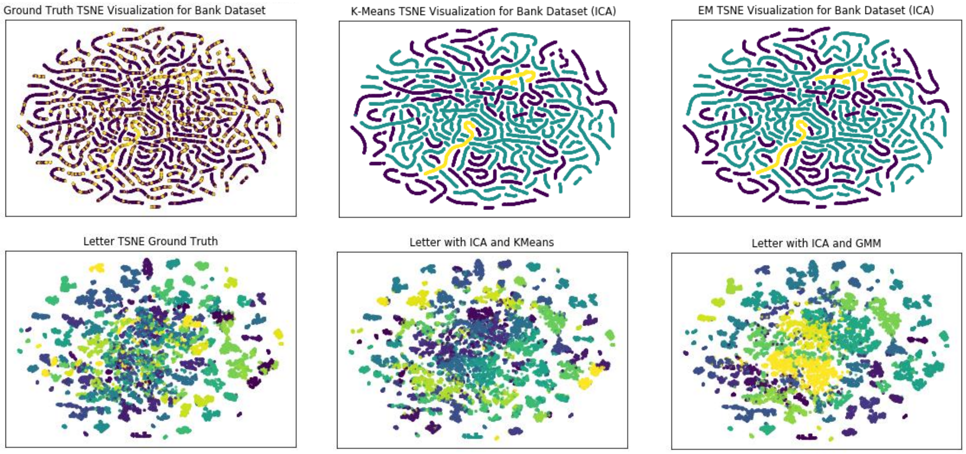


Figure 15 - T-SNE Plots for K-Means and EM with ICA Reduction

Randomized Projection (RP)

ICA and PCA are fine tools, however, they carry biases and drawbacks, such as computation costs on data with a high volume of features, and preference biases inherent to the algorithms' methodologies. Instead, randomization can be used to project a feature space into a reduced feature space. As the results of the experiments show, this is both easier and harder than using PCA and ICA and is not able to reduce the feature space by as much as the other methods above.

To evaluate RP, I used two methods using sklearn's SparseRandomProjection tool: unpruned decision tree classification (to measure noise) on the transformed data, and evaluation of reconstruction error. The randomness was controlled by using many different randomization seeds to illustrate the effect of RP over many iterations. The results of the experiment are shown in *Figure 16.* For the Bank dataset, there seems to be noticeable levels of noise in the dataset, because from 1 to 4 components, we find accuracy levels higher than when all features are included, and that these increased accuracies start to level off at 8 components. Similarly, the reconstruction error linearly decreases with some large fluctuations. With both results in mind, **13** components were used in order to ensure the important features were nearly always included in the feature set. For the Letter dataset, the results are more stable. At **9** components, however, there seems to be a plateau of performance found after the significant increases before 9, and reconstruction error approximates a linear decrease as more features are included in the projection. As such, 9 components were used for RP on the Letter dataset.

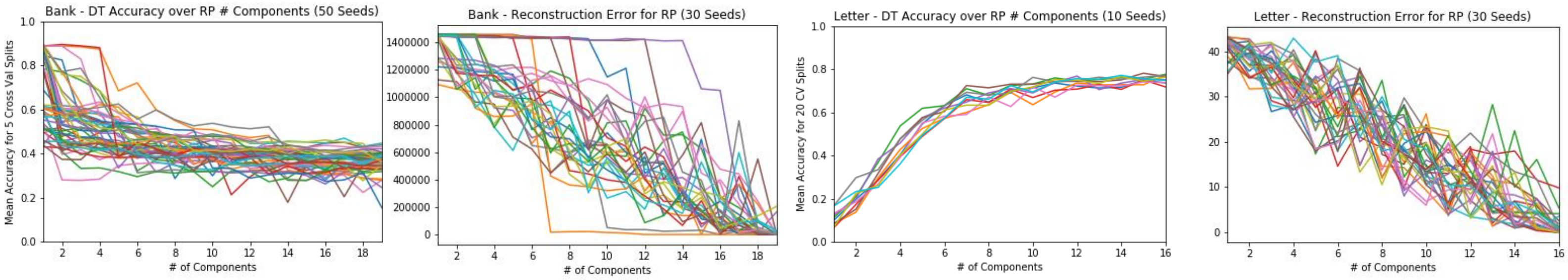


Figure 16 - Choosing the # of Components for RP

Tuning k and number of components for K-Means and EM on the Bank dataset was very similar to tuning the parameters on the full dataset. A clear elbow at a k of **4** was found in *Figure 17*, and a slight elbow was found for EM at **5** components, supported by the silhouette plot. Each plot was run with 3 different randomization seeds to control for

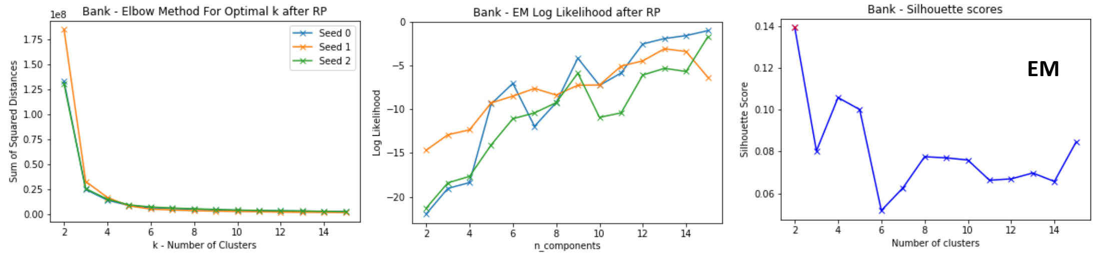


Figure 17 - Finding best # of components and k for the Bank Dataset (RP)

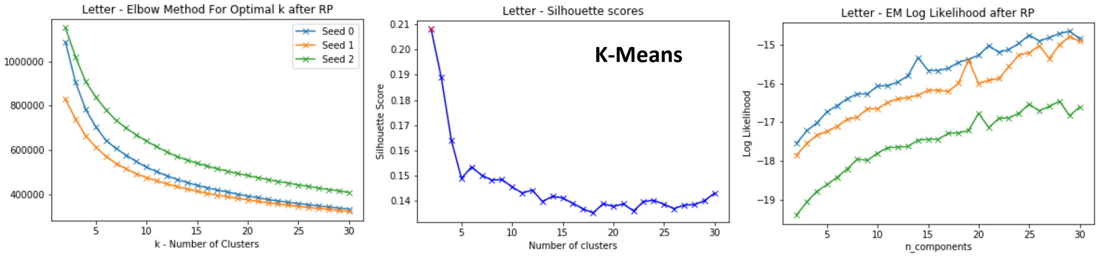


Figure 18 – Finding best # of components and k for the Letter Dataset (RP)

unexpected randomization variance, and little variance was found between each plot (i.e., they each followed a similar shape).

The Letter dataset also remained largely inconclusive, like the results from the previous experiments. A slight elbow in the distortion over k graph (leftmost image of *Figure 18*) was located at a k of **25**, which is supported as a good value on the silhouette plot. The high distortion at 2-5 make them bad candidates for k, even with high silhouette values. For EM, a plateau of the likelihood graph was also found for each run around **25** components.

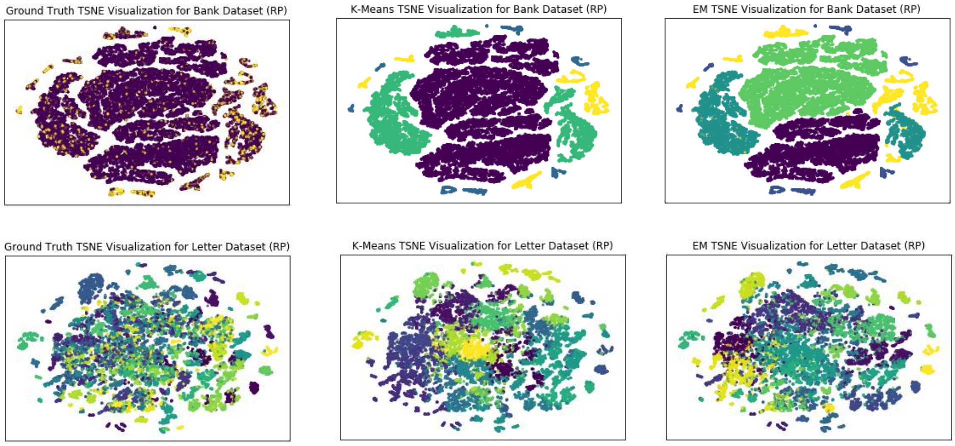


Figure 19 - T-SNE Visualization of Cluster Results (RP)

The results in *Figure 19*, for the Bank dataset remain largely like the results found in *Figure 5*, and the same sort of clustering behavior was observed after RP. For the Letter dataset, each algorithm seemed to cluster the data differently than the ground truth labelling and seems to homogenize the middle of the graph more intensely than in the previous experiments. RP makes the data even more intermingled in the Letter dataset.

RANDOM FOREST REDUCTION (RFR)

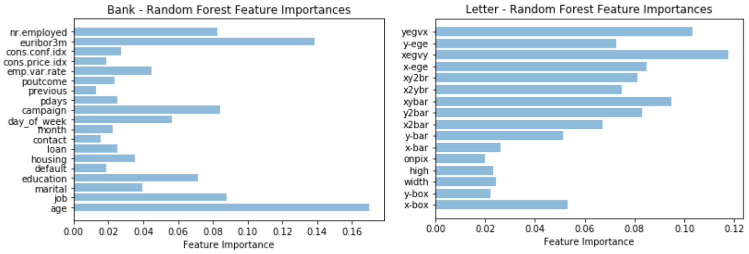


Figure 20 - RF Feature Importance

The last dimensionality reduction method used in the analysis was reduction using Random Forest feature importance. Sklearn's RandomForestClassifer, once trained on a full dataset, contains a feature importance variable that can be extracted. This requires the class labels, which are known for my datasets. The normalized importance of each feature is shown in *Figure 20*. The Bank dataset seems to have only half of the features labelled as highly important, whereas all but 5 features are given significant weight in the Letter dataset (more uniform). I used the mean importance minus a multiplier (0 for Bank, 0.5 for Letter) of the standard deviation to select the features for the reduced datasets, with a final size of **7** and **11** dimensions for Bank and Letter datasets, respectively.

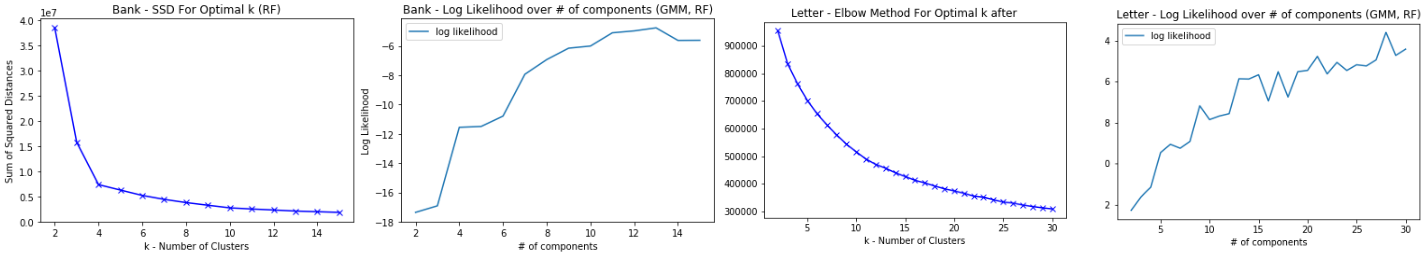


Figure 21 - Finding best k and # components for RF (Both Datasets)

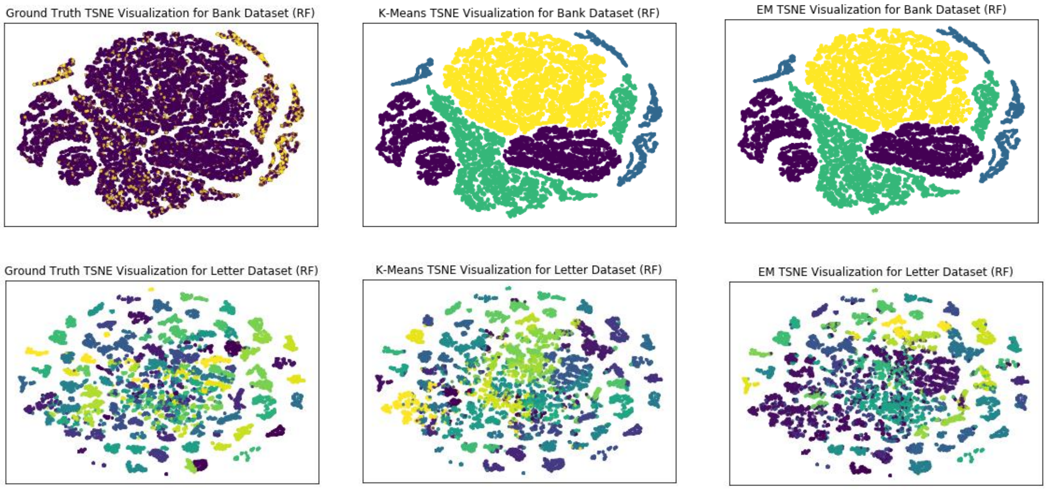


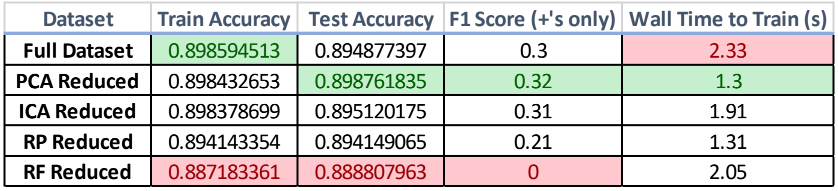
Figure 20 - T-SNE Visualization for RF Clustering Results

The results of clustering on RF reduced data in *Figure 21* show a clear elbow at **4**, supported by the silhouette plot (not shown) for both k and number of components. The Letter results had a slight elbow and likelihood spike at **27**. It is interesting that both EM and K-Means had similar 'best' k and number of components between both datasets after using RFR. Again, the clustering was plotted for visualization using T-SNE graphs, as shown in *Figure 22.* The most interesting aspect of the results for RF clustering was that the results were nearly identical for both clustering algorithms. RFR worked well for the clustering task, and it was able to significantly reduce dimensionality.

Neural Networks, Dimensionality Reduction, and Clustering (Oh My!)

Performance on Reduced Datasets

To test the effectiveness of dimensionality reduction, three components were measured against a baseline performance of the Neural Network tuned in the supervised learning analysis for the Bank Marketing Data Set: time to train, test and training accuracy, and F1 score for positive instances. F1 score on negative instances is not informative because the data consists of ~90% negative instances, so the F1 score for positive examples is more interesting of a metric. Accuracy too can be misleading, because predicting all zeros provides around 88% accuracy, depending on the cross validation split. The results are summarized in *Figure 23*.



Dataset	Train Accuracy	Test Accuracy	F1 Score (+'s only)	Wall Time to Train (s)
Full Dataset	0.898594513	0.894877397	0.3	2.33
PCA Reduced	0.898432653	0.898761835	0.32	1.3
ICA Reduced	0.898378699	0.895120175	0.31	1.91
RP Reduced	0.894143354	0.894149065	0.21	1.31
RF Reduced	0.887183361	0.888807963	0	2.05

Figure 21 - Summary of Neural Network Performance on Reduced Datasets (Bank)

The results are mostly similar, except for time to train. For each reduced dataset, time to train was reduced, and in the best case (PCA), time to train was reduced by nearly 50%. To further verify the training time improvement, a more rigorous test should be performed, however, none of the reduced datasets increased time to train, suggesting a trend of reduced time. PCA was also the method that reduced the dataset the most, so the results are within intuitive expectations.

Something very interesting about the results is the performance of the RF reduced dataset. It clearly provides the worst accuracy and F1 Score. It's possible that the features kept by the RF reduction were only predictive of negative examples, because the neural network only predicts negative class labels. This trend continues in the experiment results shown in *Figure 24* and *Figure 25*. In summary, the most important takeaway from the results in *Figure 23* is that the reduced dimensions (except for RF) did not result in significantly worse results, as originally expected. Some results were even better than the baseline. To further analyze the effect of the reduced dimensionality on the classification task, an experiment was run, this time using the cluster outputs themselves as labels.

Using Clusters as Features

The first part of this experiment used *only* the cluster labels as feature data, and the results are summarized in *Figure 24.* The results were largely the same as in *Figure 23*, surprisingly. It seems that, for this dataset, the clustering algorithms performed the task of class label retrieval very well. Some results were again even better than the baseline, such as the clusters output by K-Means with ICA reduction. Again, RF suffers the same data loss problems as in *Figure 23*, quickly

Data	Train Accuracy	Test Accuracy	F1 Score (+'s only)	Wall Time to Train (s)
Full Dataset	0.898594513	0.894877397	0.3	2.33
PCA Clusters (KMean)	0.897704281	0.895848507	0.29	1.68
PCA Clusters (EM)	0.897812188	0.894877397	0.29	3.24
ICA Clusters (KMean)	0.896841026	0.903617383	0.33	1.84
ICA Clusters (EM)	0.897677304	0.896091284	0.31	3.293
RP Clusters (KMean)	0.898001025	0.893177956	0.33	3
RP Clusters (EM)	0.897596374	0.896819616	0.27	2.03
RF Clusters (KMean)	0.887426151	0.886622967	0	0.876
RF Clusters (EM)	0.887210337	0.888565186	0	1.06

Figure 22 – Clusters as the only features

converging to weights that always predict the majority label. The time to train also did not improve from using the reduced datasets, and in fact, in many cases it took longer to train using only the clusters. For K-Means clusters after ICA reduction, test accuracy was marginally the best among test accuracy runs, while also having a relatively short time to train. It seems that ICA captures the kurtotic components of the data that avoid focusing on population level characteristics of the instances, such as age, race, and other demographics in the data that methods such as RF reduction chose (and suffered for it).

Data + Clusters	Train Accuracy	Test Accuracy	F1 Score (+'s only)
Full Dataset	0.898594513	0.894877397	0.3
PCA w/Clusters (KMean)	0.882651272	0.874969653	0.17
PCA w/Clusters (EM)	0.899349861	0.897062394	0.32
ICA w/Clusters (KMean)	0.899188001	0.897790726	0.28
ICA w/Clusters (EM)	0.899295908	0.900461277	0.32
RP w/Clusters (KMean)	0.896948933	0.897062394	0.26
RP w/Clusters (EM)	0.898675443	0.892935178	0.3
RF w/Clusters (KMean)	0.887156384	0.888565186	0
RF w/Clusters (EM)	0.888073592	0.88079631	0

Figure 25 - Clusters with the Reduced Dataset Features

For the second part of the experiment with clusters as features, the original reduced dataset was included in the feature space in combination with the clusters to observe the effect on performance. The performance either decreased or had no change, shown in *Figure 25*. It seems that, once the data is reduced for this dataset, the clustering provides the same amount of information as the reduced dataset, proving evidence that cluster results can suffice as highly reduced features in some cases.

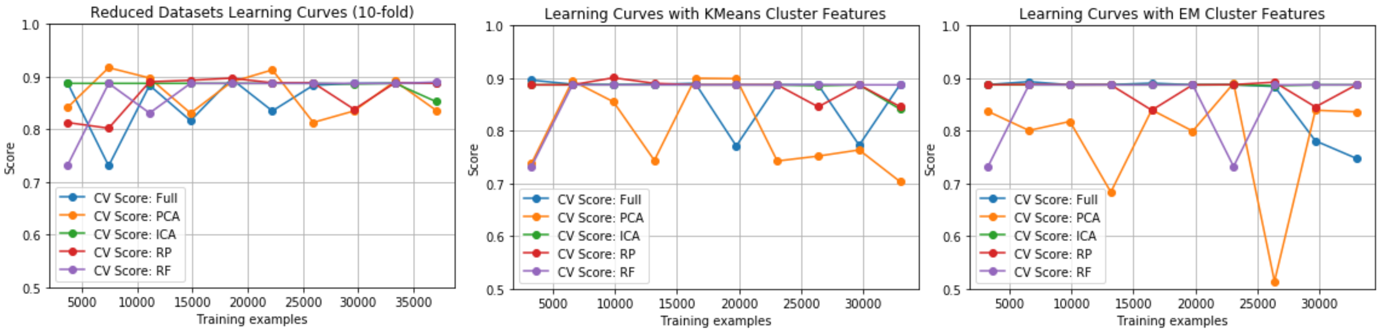


Figure 26 - Learning Curves for Reduced Data

To summarize the performance of the reduced datasets and clusters as features, three learning curves were plotted using 10 fold cross validation, shown in *Figure 26.*

The results of the learning curve show that the model's ability to generalize does suffer after most of the data reduction. In the left-most plot, the variance is not as large as the cluster-as-feature plots to the right of it. It seems the data loss at reduction and at clustering together result in inconsistent results. Interestingly, RP is the most stable of the reductions.

CLOSING REMARKS

The effect of each dimensionality reduction and clustering technique was thoroughly examined, and an argument can be made for most of the techniques in the right setting. For clustering, the Euclidean distance metric was chosen because, after experimenting with other distance metrics, there was no apparent improvement in the clustering performance. Both datasets are reduced to low dimensionality, and Euclidean distance provided the best clustering results. Despite time required to run the experiments, it would be interesting to see a more in-depth, direct comparison between different metrics. For the RF reduction, it would be also be an improvement to more thoroughly examine the selection threshold for which features to keep. The Letter dataset turned out to be a very difficult unsupervised learning problem, compared to the Bank dataset, which is opposite from the supervised learning analysis performed on these datasets.