Assignment 3: Unsupervised Learning and Dimensionality Reduction for CS7641 Fall 2021

by Jaeyong Kim

Introduction

This assignment seeks to experiment with various Unsupervised Learning Techniques as well as Dimensionality Reduction Techniques. The specific unsupervised algorithms used were K-Means Clustering and Gaussian Mixture Model (GMM) while the Dimensionality Reduction (DR) Algorithms were Principal Component Analysis (PCA), Independent Component Analysis (ICA), Randomized Projections (RP), and Kernel Principal Component Analysis (KPCA). Unsupervised Learning is very useful in the real world where ground truth labels may not be available in training the models. Dimensionality Reduction is useful for large datasets with many different features so that we can project those features into a smaller space while retaining as much information from the initial dataset as possible.

The datasets used for experiments were the same Water Potability and White Wine Quality data. Clustering was run on the datasets as well as the Dimensionality Reduction Algorithms to observe how many clusters or components could be used to describe the datasets. DR algorithms were run and clustering algorithms were applied to the algorithms to see if the performance of the clustering algorithms would change if they were passed components rather than the actual features. In addition, both the DR and Clustering algorithms were applied to the White Wine dataset and trained using the Neural Network (NN) Classifier from the previous homework.

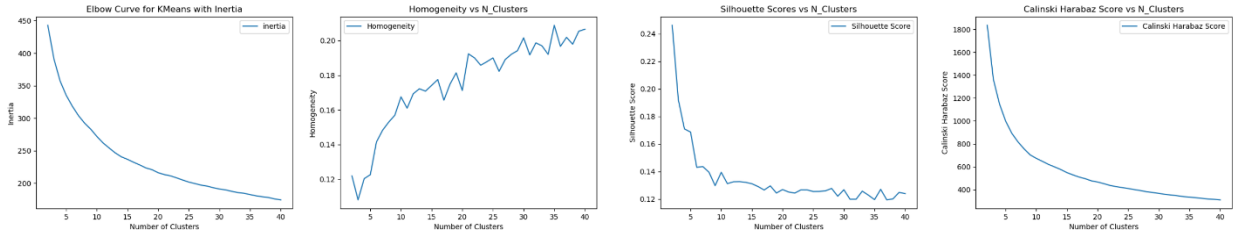
Clustering

1. K-Means Clustering

K-means clustering is a common unsupervised learning technique that is used to group data points into similar k-clusters by minimizing the variance within the cluster and increasing the distance between clusters. Inertia is the general measure used by K-Means and is measured by the sum of squared difference of a point to it's cluster center.

A. Wine Dataset

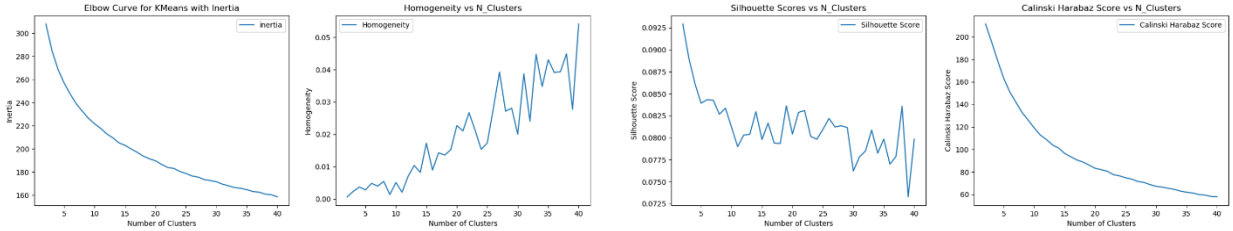
To analyze the K-Means algorithm on the wine dataset, there were various metrics used to determine what value of K might prove to be the best. The figure below shows the Inertia, Silhouette Score, Homogeneity Score and Calinski Harabasz (CH) Score as the number of clusters used increases.



In the elbow curve, it is hard to see a distinct elbow with inertia because it continuously decreases as the number of clusters increases. Therefore, to pick the most optimal k value, it is important to look at the Silhouette Score, Homogeneity, and CH Scores as the number of clusters increases. Ideally, we would pick a minimal k with high scores for Silhouette, CH and Homogeneity scores while maintaining a low value of inertia. A high silhouette score means that the points in a cluster are correctly assigned and a high CH score implies that the clusters are dense and separated from each other. Homogeneity shows us how similar the labels are within a cluster and can be useful in our case since both datasets contain labels. Though homogeneity increases as the number of clusters increase, it is not ideal to pick a k value that is high because as n_clusters approach the number of samples, homogeneity would go to 1 and it would not generalize the

data at all. We see that the silhouette score and CH score peak at k = 2 and with homogeneity, we see k = 2 is almost a local maximum. Therefore, k = 2 would be ideal for this algorithm.

B. Water Dataset

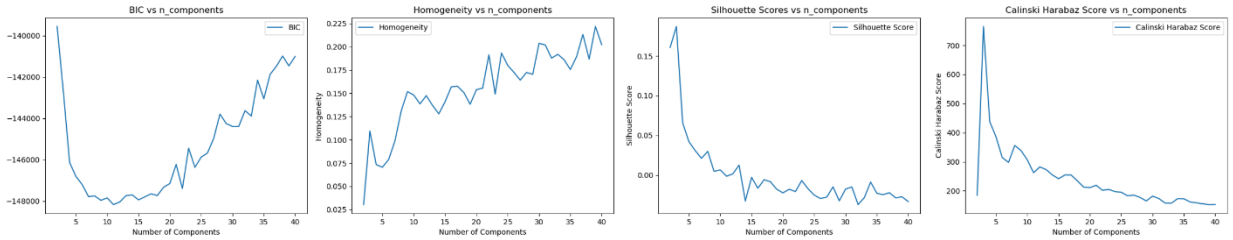


With the water potability dataset, we see a similar behavior in the elbow curve where there is no distinct elbow with inertia. Similarly, the silhouette score and CH score seem to continually decrease with an increasing number of clusters, showing us that smaller number of clusters is also better for this dataset. Homogeneity is quite low for a small number of clusters. In the silhouette chart, k = 7 shows a peak in score before a large drop off and the homogeneity is also at higher value before a slight drop off again. For this dataset, k = 7 would be the ideal number of clusters.

2. Gaussian Mixture Model

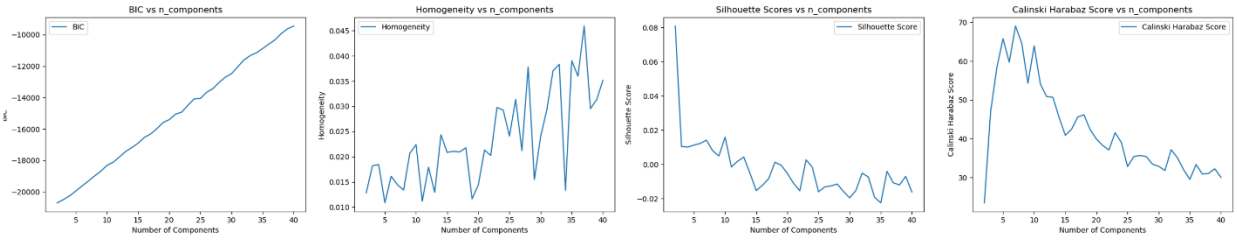
Gaussian Mixture Model is the Expectation Maximization Algorithm utilized in experimenting with theses datasets. GMM is another clustering algorithm which utilizes a mixture of gaussian distributions to predict labels for the samples within the data. It emphasizes which distribution a point is more likely to belong to.

A. Wine Dataset



For the GMM algorithm, we introduce another metric called the Bayesian Information Criterion (BIC) which measures likelihood in the model as the number of components increases in the GMM. Ideally the best model has a low BIC because BIC will penalize the number of components used as that increases. For the Wine dataset, the lowest BIC comes in around 11 components before increasing again. For the Silhouette score and CH score, we see a clear peak at 3 components. Homogeneity also has a local peak at 3 components. To fit the wine dataset, a GMM model with 3 components may be the best, although BIC shows 11 as the best number of components.

B. Water Dataset

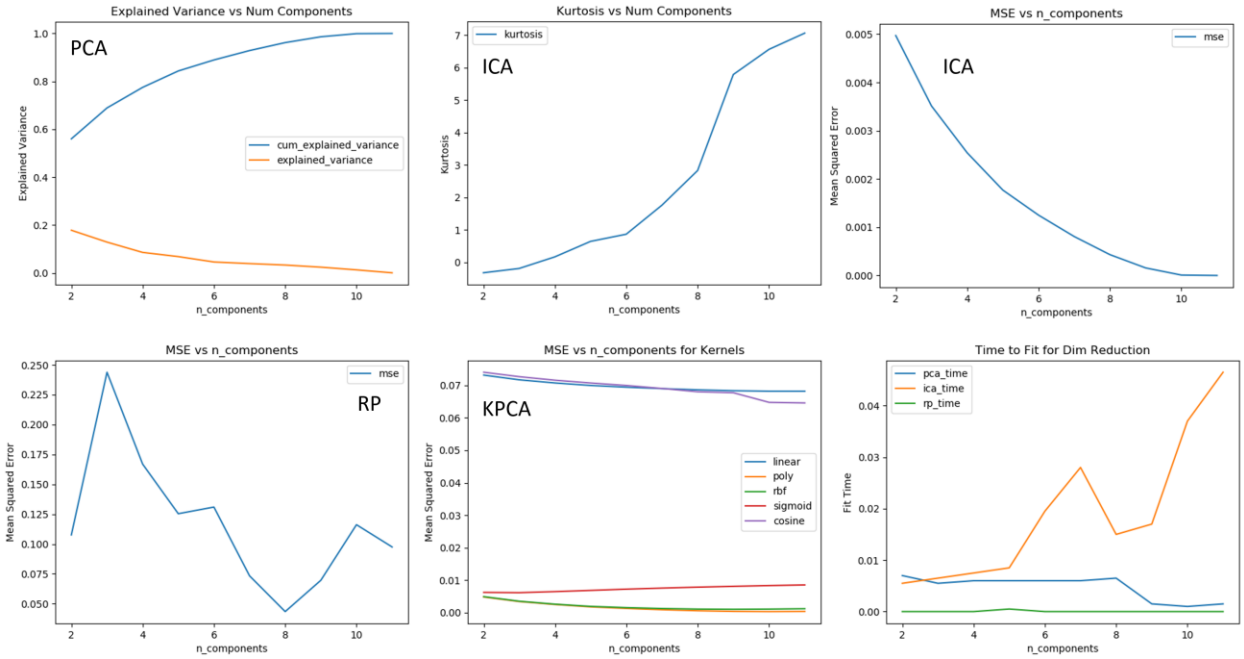


With the water dataset, the CH score has a peak at 7 components and the Silhouette score also shows a slight peak at 7 components before dropping off as the number of components grows. BIC seems to increase in a linear manner showing that utilizing the lowest number of components possible may lead to the best model and result. 7 components seem to be the best choice in this dataset for GMM.

Dimensionality Reduction

Dimensionality Reduction is particularly useful in speeding up training and fitting of various ML algorithms. Generally, when real life data comes in, there can be many features to consider, and this could cause performance issues. Dimensionality Reduction will seek to reduce these features into a smaller subspace of components while trying to retain the most information it can from the original dataset. For algorithms will be applied to the two datasets to understand how many components should be used.

1. Wine Dataset



With PCA, we see a drop off in explained variance by adding any component after the 4th component. By using 4 components, we can see the cumulative variance explained comes in right under 0.8 meaning the most important data can be captured within the first 4 eigenvectors or components.

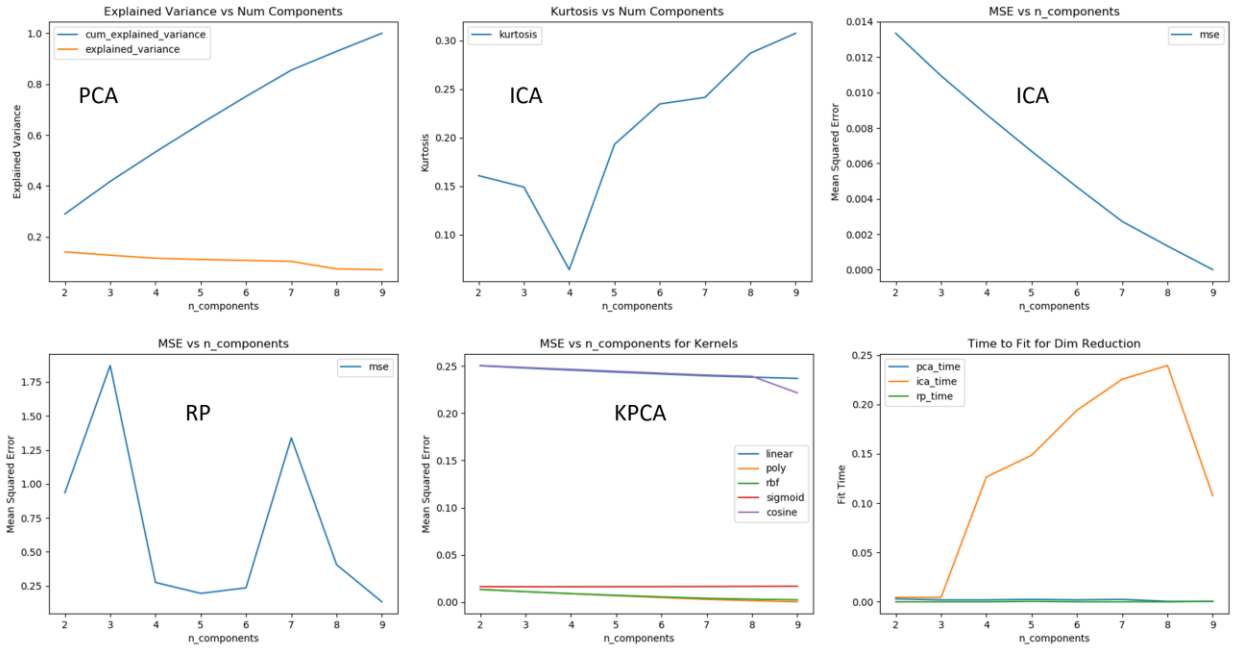
ICA shows us a different kind of transformation by taking the data and transforming it into n independent components. For ICA, we want to look at 2 things specifically, the kurtosis of adding more independent components and the reconstruction error measured by changing the number of components. Higher kurtosis is desired for ICA because lower kurtosis means a more normal distribution and more correlation between the independent components. Kurtosis of 3 means the distribution is close to a normal distribution. We also want to minimize the MSE of reconstruction and from the charts above, we see 11 components works best with ICA. The Kurtosis is higher at 11 components and shows a little bit of tapering off and there is the least MSE in reconstruction at 11 components. ICA is can be used more as a separation algorithm than a dimensionality reduction algorithm as it tries to create independent components from the features available.

Randomized Projections are known to be a faster algorithm especially compared to PCA. In this set of experiments, Random Gaussian Projections were used to compare. RP is less computationally expensive than PCA as shown in the figure above by the wall time as number of components also increases. In this case, RP uses a normal distribution to initialize it's components. For this dataset, we see 8 components for the RP algorithm achieves the lowest reconstruction error.

Another algorithm tested was Kernel PCA (KPCA). Kernel PCA is a spin off of PCA where it assumes the data is not linearly separable and operates the PCA algorithm in a higher dimension through the kernel trick. The reconstruction error shown in the previous chart shows that the polynomial kernel slightly achieves a lower reconstruction error. It also shows that the reconstruction error tapers off at 6 components.

With KPCA, the fit time took longer likely because it must transform the data first. It was not shown on the charts above because it didn't show well the scale of the other algorithms.

2. Water Dataset



For the water dataset, PCA shows a drop off going from 7 to 8 components in explained variance. Within the first 7 principal components, around 80% of the variance can be explained so we would want to decompose the dataset to its first 7 components.

ICA shows the highest kurtosis coming with 9 components which is how many features this data set includes. The reconstruction error shows the lowest value at 9 independent components. One thing to note is that the kurtosis values measured in this dataset is significantly less than the in the wine dataset. This likely means that there is a lot of correlation between the independent components. In this case, taking the 9 independent components helps capture the original data the best.

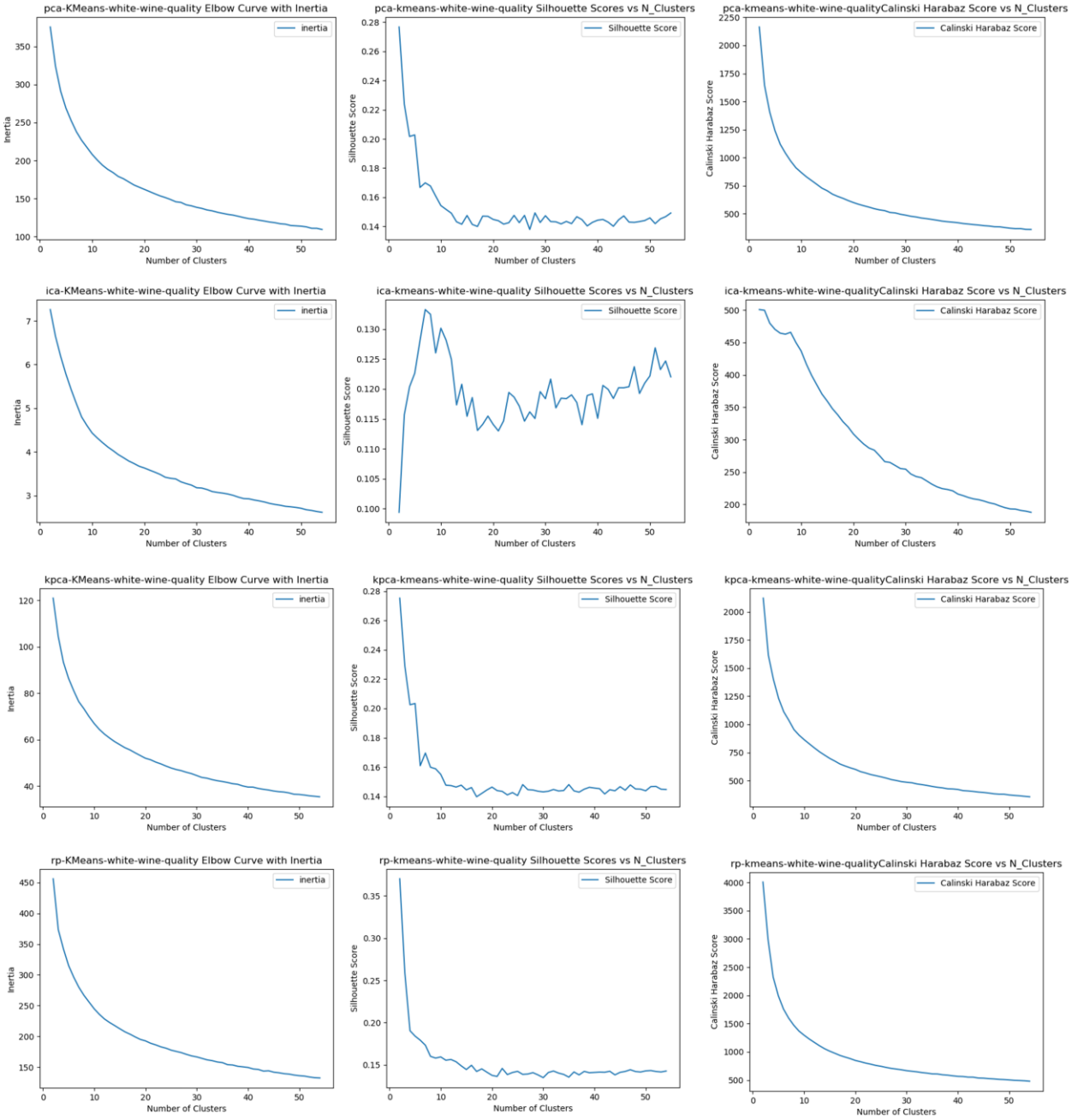
RP shows optimal number of components to be 5 because the reconstruction error seems to be at a minimum. 9 components also show a minimum for the reconstruction error, but for the sake of dimension reduction, we want the lower number of components. In addition, RP performs slightly faster than the other algorithms and consistently takes the same amount of time as the number of components increases.

The kernel PCA algorithm shows prominent performance with the polynomial kernel and the error tapers off at around 7 components.

Dimensionality Reduction and Clustering

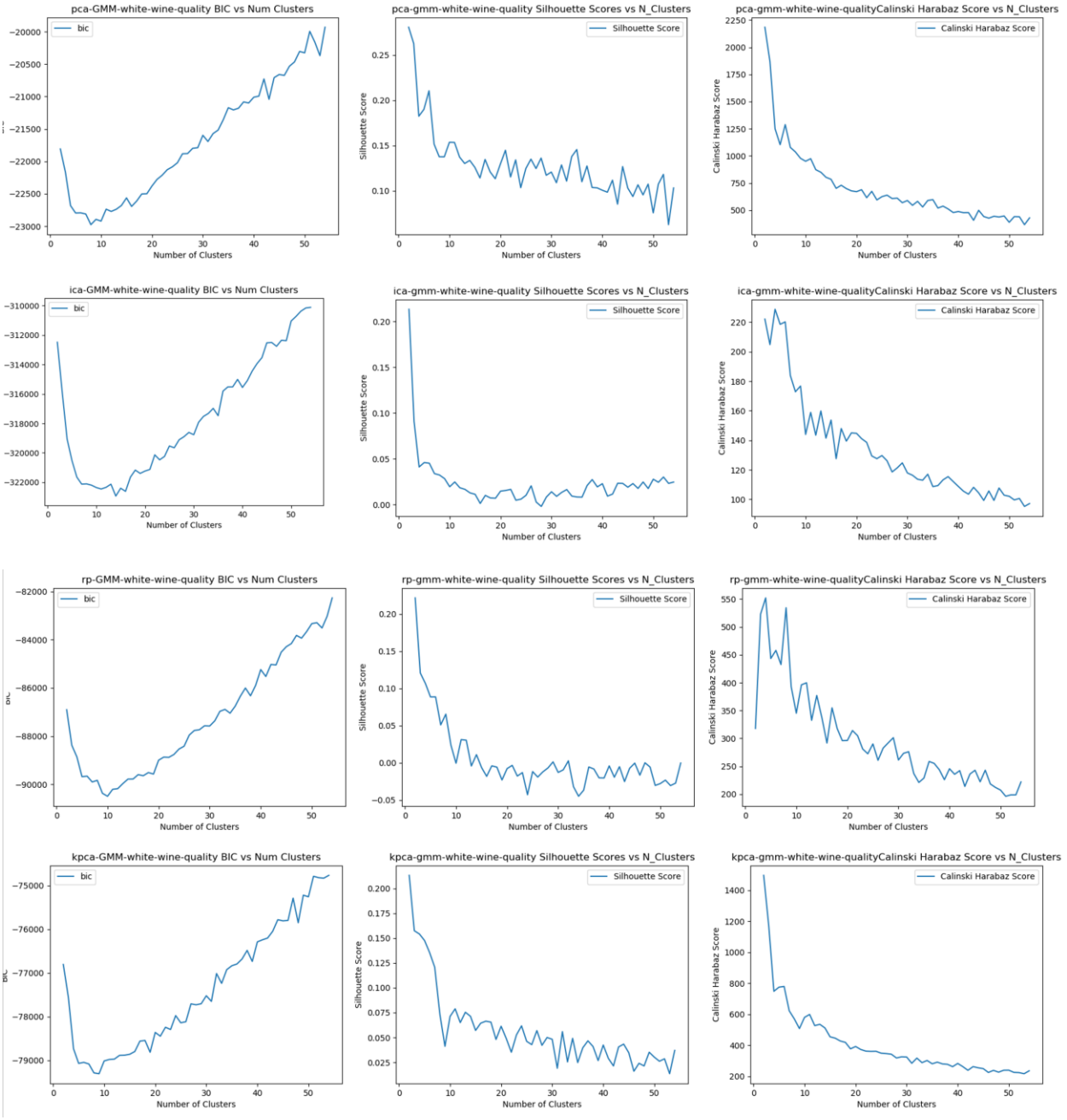
This section will discuss the observations from reducing the original datasets and running clustering algorithms on the reduced datasets. The parameters for reduction were taken from the previous experiments.

1. Wine Dataset



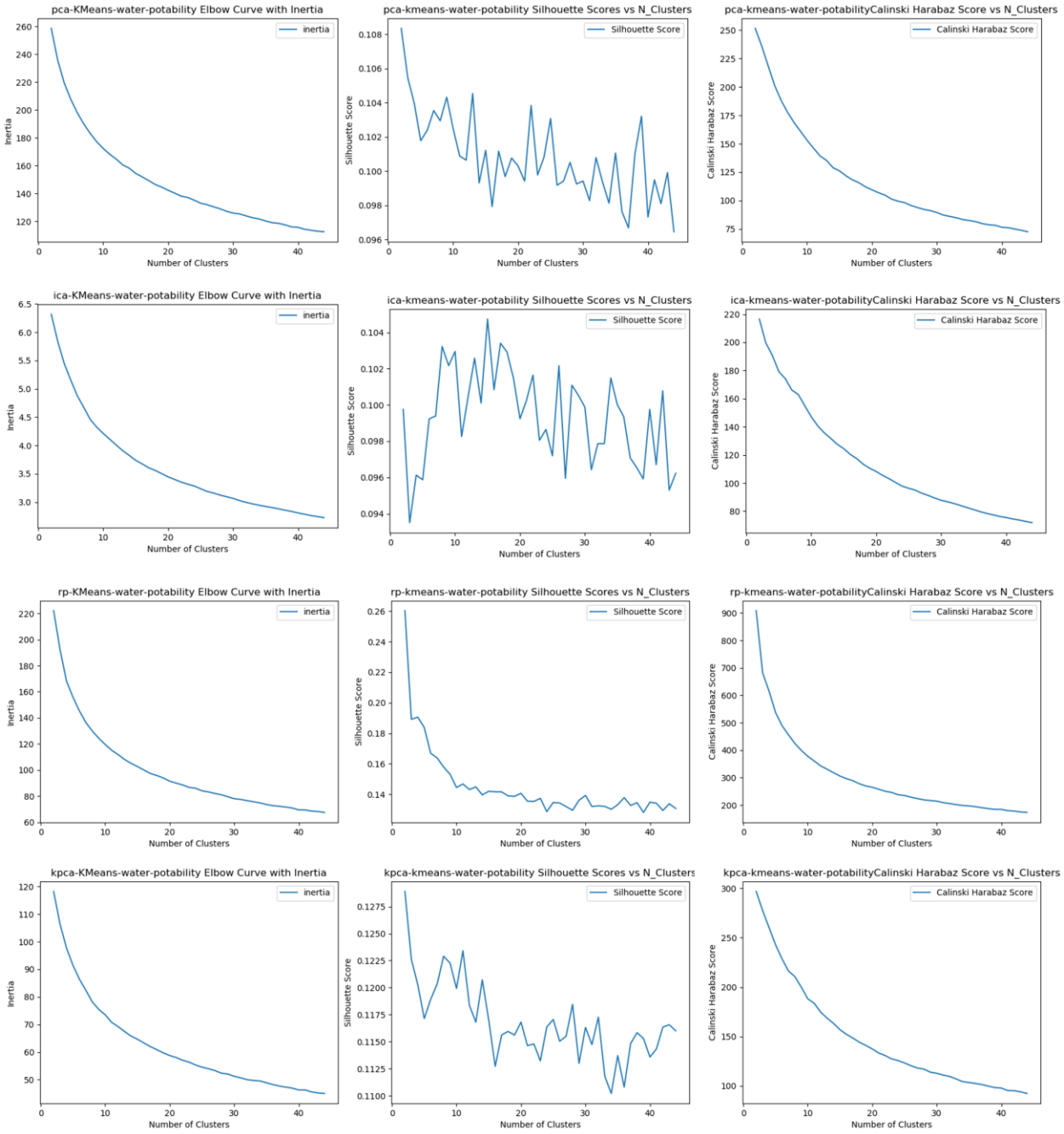
The charts above show the elbow curve with Inertia, Silhouette Scores and CH scores after each DR algorithm was run on the K-Means clustering algorithm. The general trend is that PCA, KPCA, and RP generally indicate an ideal number of clusters as 2. However, with ICA there is a slightly different trend where the ideal number of clusters seem to be around 10 clusters. In addition, the scale of inertia measured with the ICA and K-Means pairing is significantly less than any of the other algorithms by a power of 10. This low inertia means that clusters created from ICA are much closer to the respective centers and with ICA, K-Means creates more clusters, but due to the lower Silhouette and CH scores, these clusters from ICA may not be as easily separable. It is also observed that RP with 8 components maintains higher levels of Silhouette Scores and CH scores giving tighter and more separable clusters. Compared to the k-means

clustering on the raw wine data, RP actually gives more distinct and separable clusters also shown by the higher silhouette and CH scores.

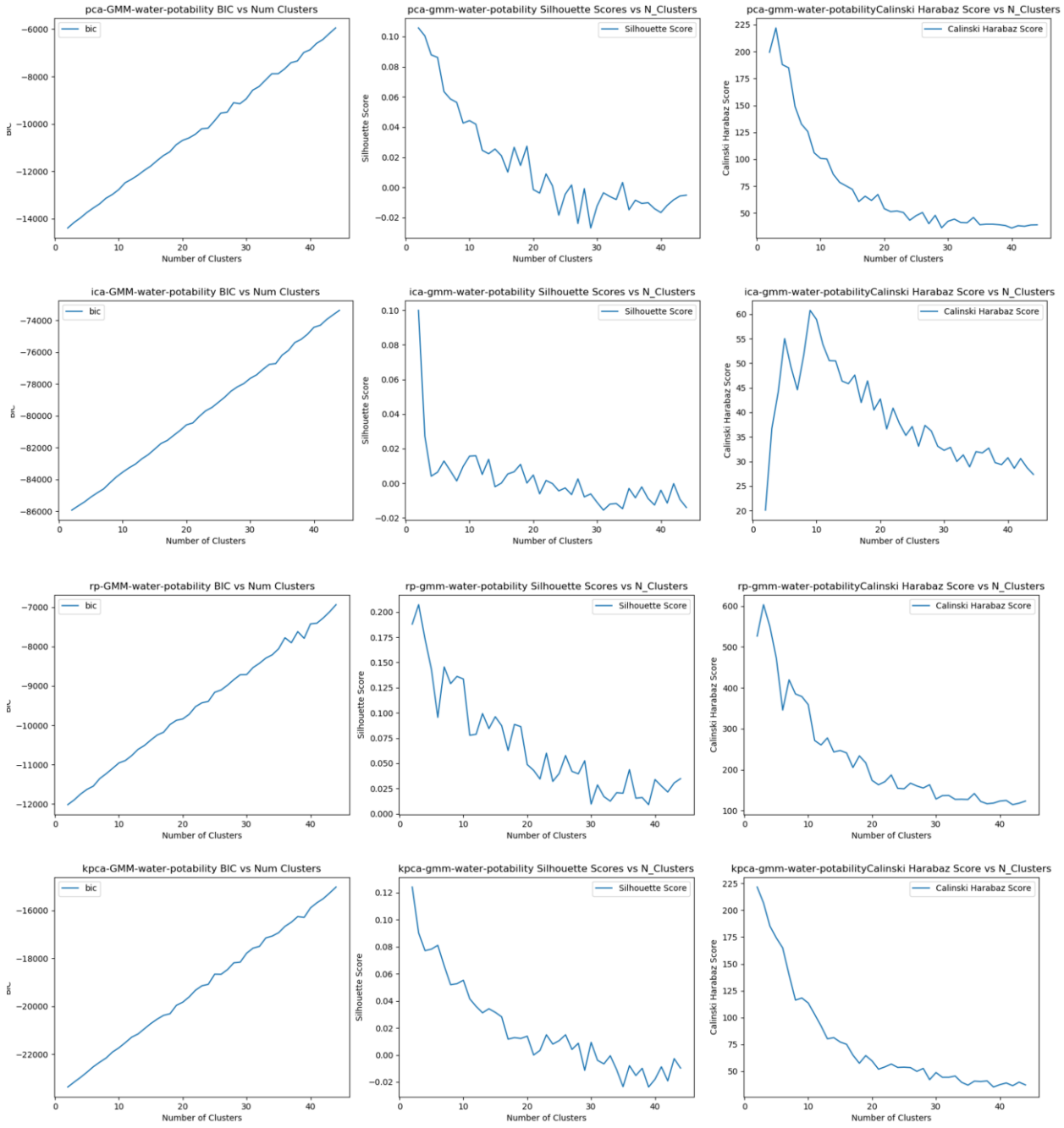


The figures above show the performance of dimensionality reduction on the GMM algorithm for the wine dataset. The BIC scores all show optimal cluster numbers being around 9-10 clusters. The CH scores and the Silhouette scores tend to show lower number of clusters are a better fit, but also support the low BIC score around 9-10 clusters with local peaks at the number. It seems like PCA would have the best clusters at 10 clusters because it has the highest Silhouette score and highest CH score. Similar to k-means, applying the PCA to the data before GMM gives better clustering performance as well.

2. Water Potability



Similar results can be seen here where ICA seems to show k to be around 15 clusters with the newly transformed data. It seems like with the independent components from ICA, K-Means can make more smaller clusters especially with the inertia values being so low compared to the other algorithms. With the other reduction algorithms, there is still a similar pattern of creating smaller number of clusters with each point a larger distance from the center. With this dataset, Random Projections show highest Silhouette score and CH score for 2 clusters which mean the clusters are tighter and more separable. Running dimensionality reduction shows significant improvement over running the clustering algorithms on the regular dataset and improves the separability and tightness of the clusters.



Like the GMM algorithm on the pure datasets, BIC shows that any model with a lower number of components will be the better fitting model to use. PCA shows a peak in CH score at 3 components, and it seems like Silhouette score decreases as the number of components decreases. RP and KPCA also validate this showing peak Silhouette and CH scores around 2 to 3 components. RP shows a higher silhouette and CH scores meaning that the combination of RP with GMM produces the best dense and separable clusters compared to the other reduction algorithms.

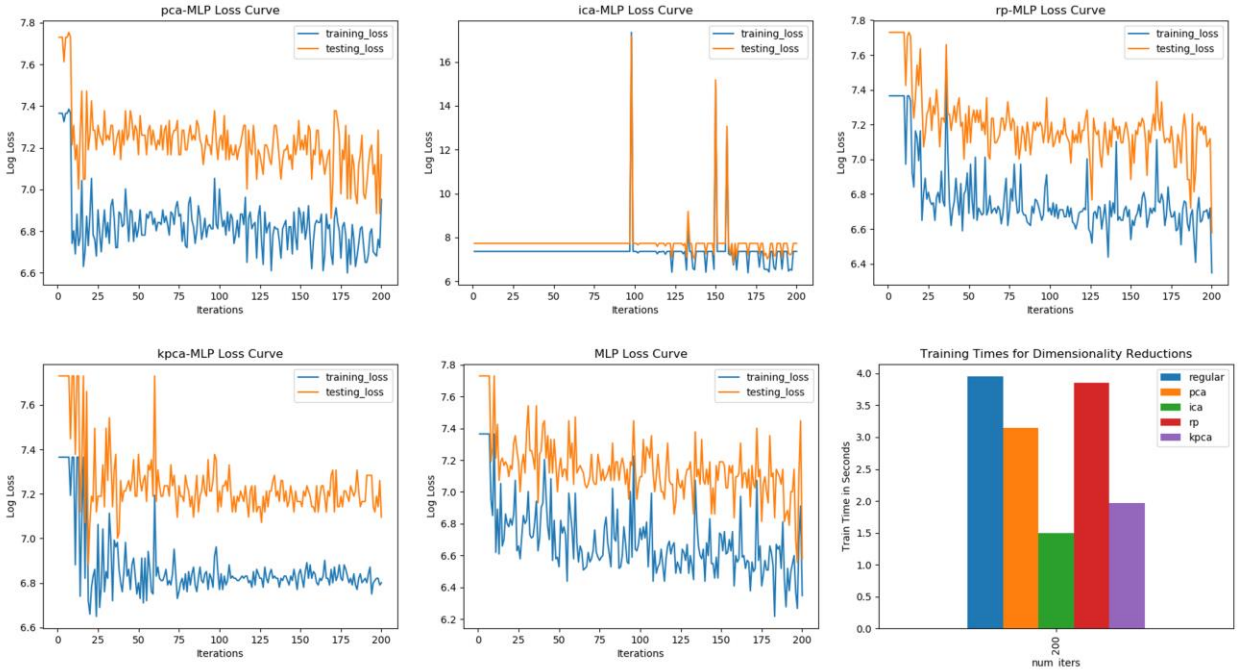
3. Overall

From the experiments above, it seems like RP and PCA generally perform similarly on each dataset with KPCA and ICA generally lagging in terms of silhouette and CH scores. From previous experiments with pure ICA, we found that the best ICA performance comes from a higher number of independent

components. With the addition of clustering, ICA creates a higher number of clusters for both clustering algorithms that seem to be tighter from the small inertia values. This makes sense since ICA works to create independent components from the initial data. RP can be the more favorable of the algorithms because it is less computationally expensive yet can keep up in minimizing the reconstruction errors. In most cases with both k-means and GMM, applying PCA or RP helped the clustering algorithms output tighter and more separable clusters evidenced by the improved silhouette and CH scores. Another thing to note with PCA and KPCA is that there is not much performance difference in clustering after KPCA is applied to the dataset. This could mean the original data is can actually be decomposed into a linear subspace and the kernel method is not actually necessary.

Dimensionality Reduction and Neural Network

Dimensionality reduction is often used with another algorithm to improve the performance. This can surely be the case with computationally expensive learners such as neural networks. For these experiments, the wine dataset was analyzed after undergoing dimensionality reduction with each of the different algorithms.

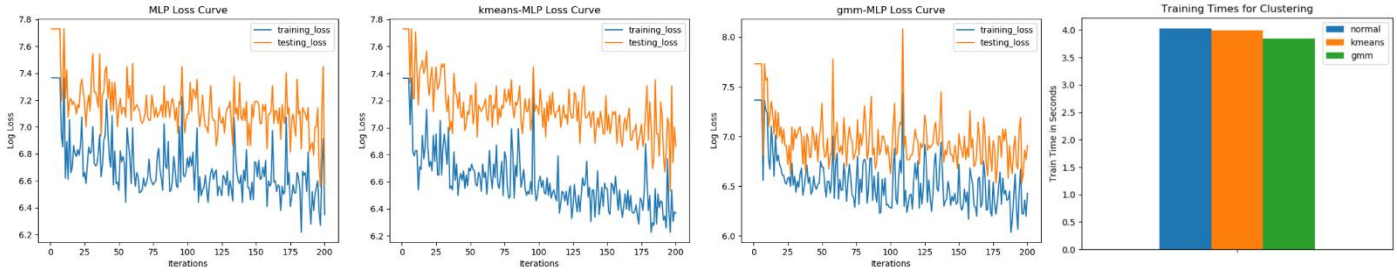


Starting with the wall time comparison, training over the reduced datasets was significantly less than the normal dataset over 11 features. The exception was RP which is interesting because the chosen number of components for RP was 8 components while ICA used 11 components. From the Loss Curve, it does not seem like ICA decomposition is not very helpful with training the neural network because many iterations show a flat loss curve. This could be why ICA training time was much less. According to the loss curves, PCA and KPCA generally run the same achieving a loss of around 7.1-7.2 after 200 iterations of training. RP achieves a similar loss of 6.6 after 200 iterations as the regular Neural Network classifier over the original dataset. This shows that using dimensionality reduction could have similar performance with less features compared to the normal dataset. PCA and KPCA also achieve great performance with less components than both the regular MLP and the RP applied MLP learner. KPCA training time is 50% less than the normal MLP and the PCA classifier having a training time about 25% less than the regular classifier. PCA can be seen as an effective dimensionality reduction algorithm as it reduces the training

time over the learner while still maintaining decent accuracy and performance over training with the whole dataset.

Clustering and Neural Network

Clustering can also be used to improve models by using the predicted and fitted clusters as extra features for the models. For this experiment, the clusters from running GMM and K-means on the wine dataset are passed in as inputs to the neural network classifier to see if it would improve the performance.



The figure above is the result of adding the clustering information as extra features in the dataset to train the neural network. 2 clusters were used with K-Means and 3 clusters/distributions were used for the GMM algorithm. There is only slight variation in the wall time to train over 200 iterations with GMM clustering and NN training being marginally faster. In terms of model performance, the K-Means clustering has the highest loss out of the three algorithms and GMM also performs marginally better seemingly achieving a lower testing loss than the regular NN and K-means classifier.

Using clustering to add features to the data does not have any major impact on improving the learning ability and performance of the NN classifier. Loss results in being around the same value of about 7.0 and the training times are not much better either. It would be more interesting to see the performance of the neural network learner after both dimensionality reduction and clustering are applied to the database. From the previous clustering and reduction experiments, we can see that applying the dimensionality reduction to the dataset before clustering resulted in more distinct clusters and that could provide more value to the neural network learner.

Conclusion

The above experiments sought to understand Unsupervised Learning and Dimensionality Reduction algorithms and how they could be utilized to improve performance of a variety of algorithms. With clustering, to find the ideal clusters, we analyze inertia, silhouette scores and CH scores to understand the structure of the clusters. Low inertia paired with high silhouette and CH scores provides the optimal shape of clusters. Clustering performance can also be improved by applying dimensionality reduction to the data first. While the data used here are not high dimensionality data by any means, the experiments show that DR algorithms such as PCA and RP can take high dimensionality data and project it into lower dimensions while maintaining performance and accuracy. Both PCA and RP were adept at reducing the correlated data into principal components or independent components in the case of the ICA algorithm. Dimensionality reduction also performed well when used prior to training a MLP Classifier. Specifically, PCA, KPCA and RP can improve the time it takes to train a classifier and still maintain similar accuracy and performance. While the same can not be said for clustering, perhaps applying dimensionality reduction and clustering to a dataset prior to training a classifier could be beneficial and could be pursued further in more experiments.