Project3 Report - Unsupervised Learning and Dimensionality Reduction

Kyle Grace
kgrace6@gatech.edu

Abstract—Unsupervised learning is the process of taking a dataset with no labels and reducing it to a series of subsets of data. The purpose in doing this is to make order in the data even if there is not a predetermined method of organization. Dimensionality reduction is the process of taking the dataset and simplifying it based on the features it contains. This can be done several ways, but the idea is to reduce the complexity of the inputs such that the least information is lost while optimizing the efficiency of the learner.

1 DATASETS

The same 2 datasets from Project 1 are being used for testing Unsupervised Learning and Dimensionality Reduction methods, and therefore, the reasons they are interesting are largely the same, but it will be reiterated here so that report 1 is not needed to understand this one.

The first dataset is a water quality model from kaggle. This dataset is a numerical dataset that associates several metrics with water "potability", which is 0 if the water is unsafe for human consumption, or 1 if it is safe. There are 9 features: ph, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic carbon, Trihalomethanes, and Turbidity, all of which are floating point values.

There are a few factors that make this dataset interesting. The primary reason is that even though it is just a binary classifier, it is still surprisingly challenging to get accurate predictions. One might think that a binary classifier with 9 available features would get very high accuracy models, but that hasn't been the case. At the same time, it seems to have good quality data and shows varying qualities in different models. The features are also continuous, which likely factors into the difficulty.

The second dataset is also a binary classification model showing the likelihood of heart attack based on 13 health metrics. The 13 health metrics are: age, sex (0 or 1), cp (chest pain of 4 types), trtbps (resting blood pressure), chol (cholesterol), fbs (fasting blood sugar), restecg (resting electrocardiographic results), thalach (max heart rate), exng (exercise induced angina, 1 for yes, 0 for no), oldpeak (the previous peak), slp (slope), caa (number of major vessels), and thall (Thal rate).

Even though at a surface level this looks similar to the water quality dataset, there are some very interested differences. The two main factors that make this dataset different from the water potability dataset is that the heart data is more categorical, meaning many of the features, although represented as numbers, are actually putting the feature into a finite set of groups, where the water potability data is all continuous. It also means that "distance" metrics may not perform the same way since categorical features converted to numbers don't necessarily maintain any significance when the numbers are far apart. As a result.

Lastly, the water dataset is roughly an order of magnitude larger than the heart dataset. The difference in size should play into how well different learners perform. However, even though it is smaller, it has more features, and many of them are categorical, which means the complexity is reduced. It should be interesting to see how those factors play against each other.

2 CLUSTERING ALGORITHMS

The two clustering algorithms used in this experiment are K-Means and Expectation Maximization.

2.1 K-Means Clustering

K-Means Clustering is a method which groups, or clusters, of data are associated together by mean values. Essentially, a number of clusters, k, is chosen, and then the input data is grouped into that many clusters. Each data point is associated with the cluster with the nearest mean value. It is an iterative process whereby the end result should be k clusters which are filled with similar data, or at least that's the goal.

2.2 Expectation Maximization

Expectation Maximization is an iterative method of clustering which attempts to find maximum likelihood estimates of parameters. It gets its name from alternating steps of "expectation", which creates evaluates likelihood based on the currents parameter estimates, and "maximization", which computes the parameters which will maximize the next expectation step. This has been implemented in the form of a Gaussian Mixture model, which uses Gaussian distributions for the estimation step. The end result is somewhat similar to K-Means clustering in that there should be a specific number of clusters, or components, each with similar data inside the group, but uses a different process to get there.

3 CLUSTERING RESULTS

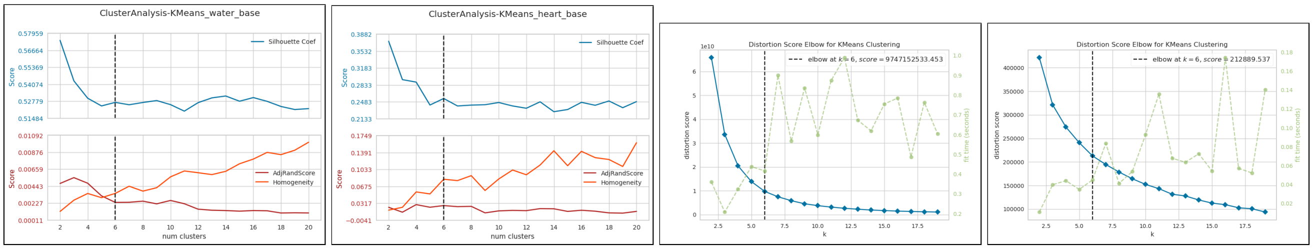
Each clustering method is shown here and analyzed using Silhouette score, Homogeneity score, Adjusted Random Score, and an elbow graph. The Silhouette score is a measurement of how well clusters are separated from each other. A value of 1 means that clusters are well separated and clearly distinguished. A value of 0 means that they are indifferent and there is no separation. Homogeneity is a measurement of how much data point within a cluster are from the same class. A high Homogeneity score means that most items within each cluster have the same label. The Adjusted Random Score is a metric within Scikit-Learn defined as follows: "The Rand Index computes a similarity measure between two clusterings by considering all pairs of samples and counting pairs that are assigned in the same or different clusters in the predicted and true clusterings." It is a similar measurement to Homogeneity in that it compares the predictions with true labels, and is low for random labels and high for well defined (non-random) labels.

Lastly is the Elbow graph. This method shows distance measurement on the y-axis, and number of clusters on the x. The distance can be calculated in a number of ways, but for this project I have chosen to use distortion. Distortion measures Euclidean distance between points, or rather the Euclidean difference in value for each feature. While there are other possible ways to measure the distance, distortion is a relatively simple method which performs well, and will also illustrate some of the differences in performance when comparing continuous and categorical data. As mentioned before, when converting categorical information into numerical representations, the "distance" is arbitrary, but will be taken into consideration using this method.

A good number of clusters should have higher values for Silhouette, Homogeneity, and Adjusted Random Score. The Elbow method attempts to find the point in the Elbow graph where the distance most sharply levels out. In other words, the point where an increase in k starts to have the smallest improvement on distance.

3.1 K-Means Results

As can be seen from figure 1, the number of clusters chosen for K-Means selected for the Water dataset is 6, and for the Heart dataset is also 6. Homogeneity steadily increases with higher k, which makes sense considering smaller clusters are more likely to contain similar labels as long as there is a reasonable method for sorting. However, the Adjusted Random score slowly reduces, and the Silhouette score sort of levels off. Combining each of these metrics and the Elbow graph, it can be seen that the "optimal" number of clusters comes at points where there are subtle



(a)

(b)

(c)

(d)

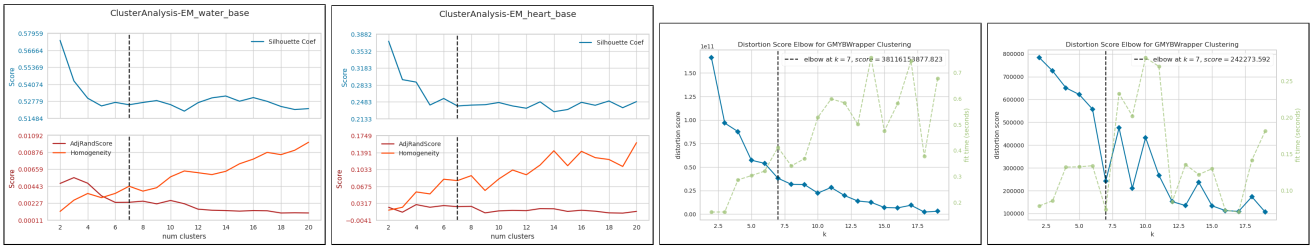
Figure 1—Cluster Analysis for KMeans models. (a) Water Dataset Silhouette, Homogeneity, and Adjusted Rand Score (b) Heart Dataset Silhouette, Homogeneity, and Adjusted Rand Score (c) Water Dataset Elbow Graph (d) Heart Dataset Elbow Graph

peaks or leveling regions in each metric.

For example, looking at the Water dataset, there is a small peak for the Silhouette score, a leveling off of the Adjusted Random score, and a midpoint for Homogeneity. Choosing 4 clusters would have been another good choice, but it's very challenging to say for certain which is objectively better considering all the factors at play. The Heart dataset has more defined peaks and elbows, but there still is not a very obvious and absolute best choice, so this needs to be taken into consideration in future sections when this same analysis is compared when using Dimensionality Reduction.

The clusters themselves are somewhat hard to display or describe. Since the input datasets both only has 2 possible labels, a high number of clusters doesn't necessarily mean anything as it relates to the input labels. However, these are captured in the metrics such as Homogeneity and Adjusted Random score. The clusters managed to have positive values in both metrics for both datasets, but not very high values. It seems that the clusters that make for more distinct and separable groups don't necessarily line up with the labels that are desired.

3.2 Expectation Maximization Results



(a)

(b)

(c)

(d)

Figure 2—Cluster Analysis for Expectation Maximization models. (a) Water Dataset Silhouette, Homogeneity, and Adjusted Rand Score (b) Heart Dataset Silhouette, Homogeneity, and Adjusted Rand Score (c) Water Dataset Elbow Graph (d) Heart Dataset Elbow Graph

Expectation Maximization looks very similar to K-Means, as it should. The concepts are very similar, with different approaches to get the organized data. Strangely, even though the same function were used to determine the "optimal" k value, and all of the metrics follow very similar patterns, the k values selected were different. Keep in mind that the plots and k values were all automated in the code, so they may not reflect the actual best value.

For the Water dataset, the best k value chosen was 7, likely because of the peak in Homogeneity, whil still maintaining a decent Adjusted Random score. There is not really an obvious "elbow", and are more pronounced at 5 and 8, but both of those choices are less ideal in Homogeneity, and comparable in both Silhouette and Adjusted Random score.

Any of those values would probably be fine.

For the heart dataset, the elbow is much more pronounced, but all of the other metrics are less ideal than at k=6. 13 is another option which has a distinct elbow, and still has strong values for each of the other metrics. This goes to show that there is not necessarily a single selection that is perfect. For the purposes of analysis, I will stick with the elbow selection method when comparing methods just so that there is a consistent measurement between experiments.

Similar to K-Means, the clusters themselves are not very interesting. Unsupervised learning is about separating the data into groups that are more independent, and different from each other, while the labeled data is looking for some specific attribute which can be determined by the features present. Often, this is not an easy task, otherwise machine learning would not even be necessary. So, the clusters generated don't exactly match anything tangible or real, at least in the case of these datasets.

4 DIMENSIONALITY REDUCTION METHODS

There are 4 methods for Dimensionality reduction tested here: PCA (Principal Component Analysis), ICA (Independent Component Analysis), RP (Randomized Projections), and NMF (Non-Negative Matrix Factorization).

4.1 Principal Component Analysis

PCA is a method of data reduction that transforms high dimensions into lower dimensions while retaining as much information as possible. This is accomplished by linearly transforming data into a new coordinate system such that the greatest variance lies in the principal component, and the rest in reducing order.

4.1.1 *Results*

The best number of components was selected by measuring the cross validation accuracy as well as the variance. Since PCA is fundamentally connected to variance, one way to select principal components is by finding the number of components where variance stops changing by a significant amount, and similarly where the cross validation score is close to maximum while still maintaining a relatively low number of components. A low number of components is desired simply because the entire purpose of this exercise is to simplify the dataset and make it more efficient.

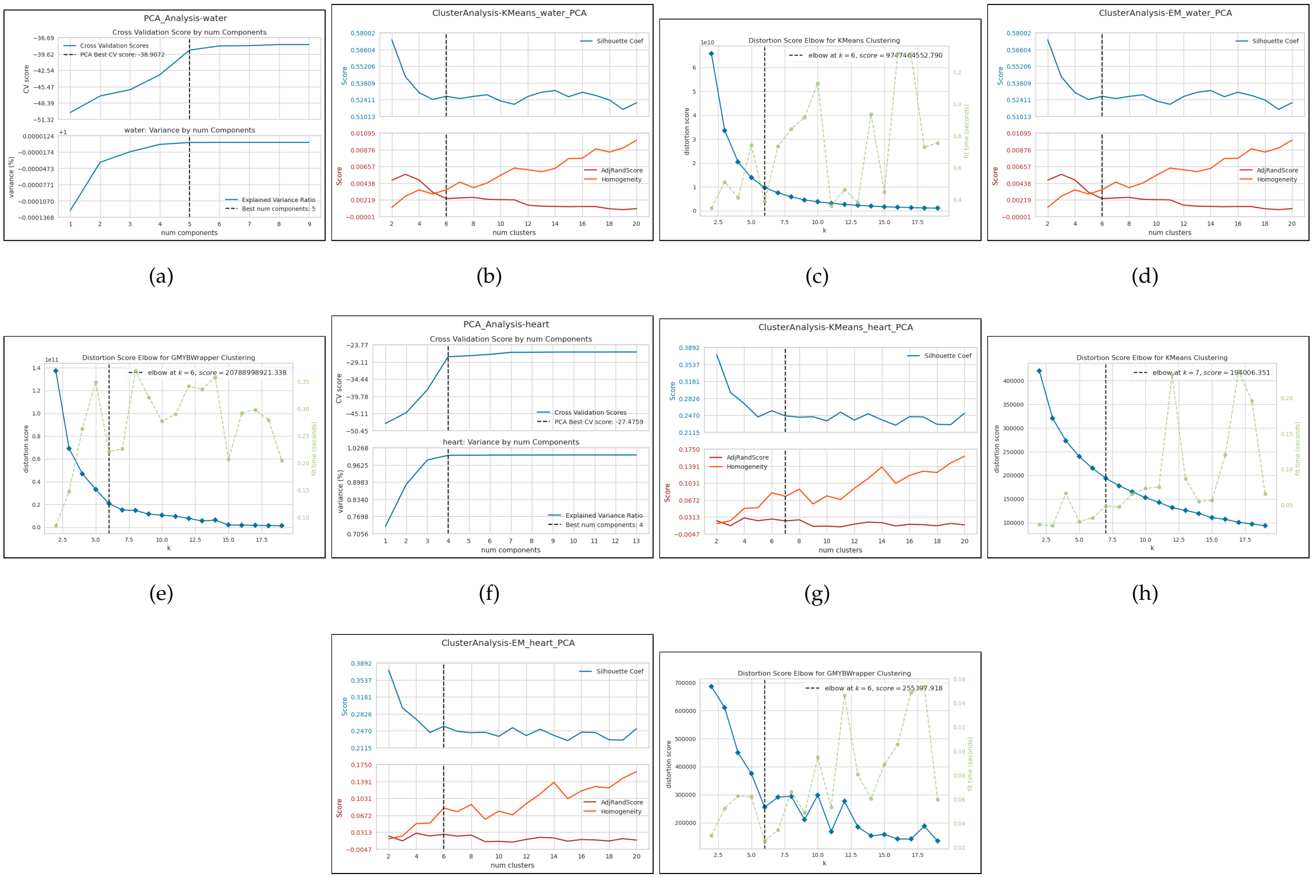
Both the Water and Heart datasets have sharp turns in cross validation score at the same point where the variance levels off. For Water that is at n=5, and for heart that is at n=4. If you take a look at the metrics for each dataset, they are largely unchanged in both cases. There is some slight variation in the optimal number of clusters selected, but as discussed previously, that is not an exact science anyway.

4.2 Independent Component Analysis

ICA is another method of dimensionality reduction which focuses on figuring out which components, or features, are independent of each other. It differs from PCA because PCA focuses on compressing the information (with minimal loss), and ICA focuses on separating information into independent sets.

4.2.1 *Results*

ICA uses kurtosis to measure what the best selection is. Kurtosis is a measurement of how pointed the Gaussian distribution is. In other words, high levels of kurtosis means a higher concentration of points near the median. Since ICA focuses on independence of features in a dataset, high kurtosis means high correlation for each component, and lower associated with the other components. Both datasets resulted with peak kurtosis values over 0.8, which is fairly significant. They managed to find components that have high measurements of independence.

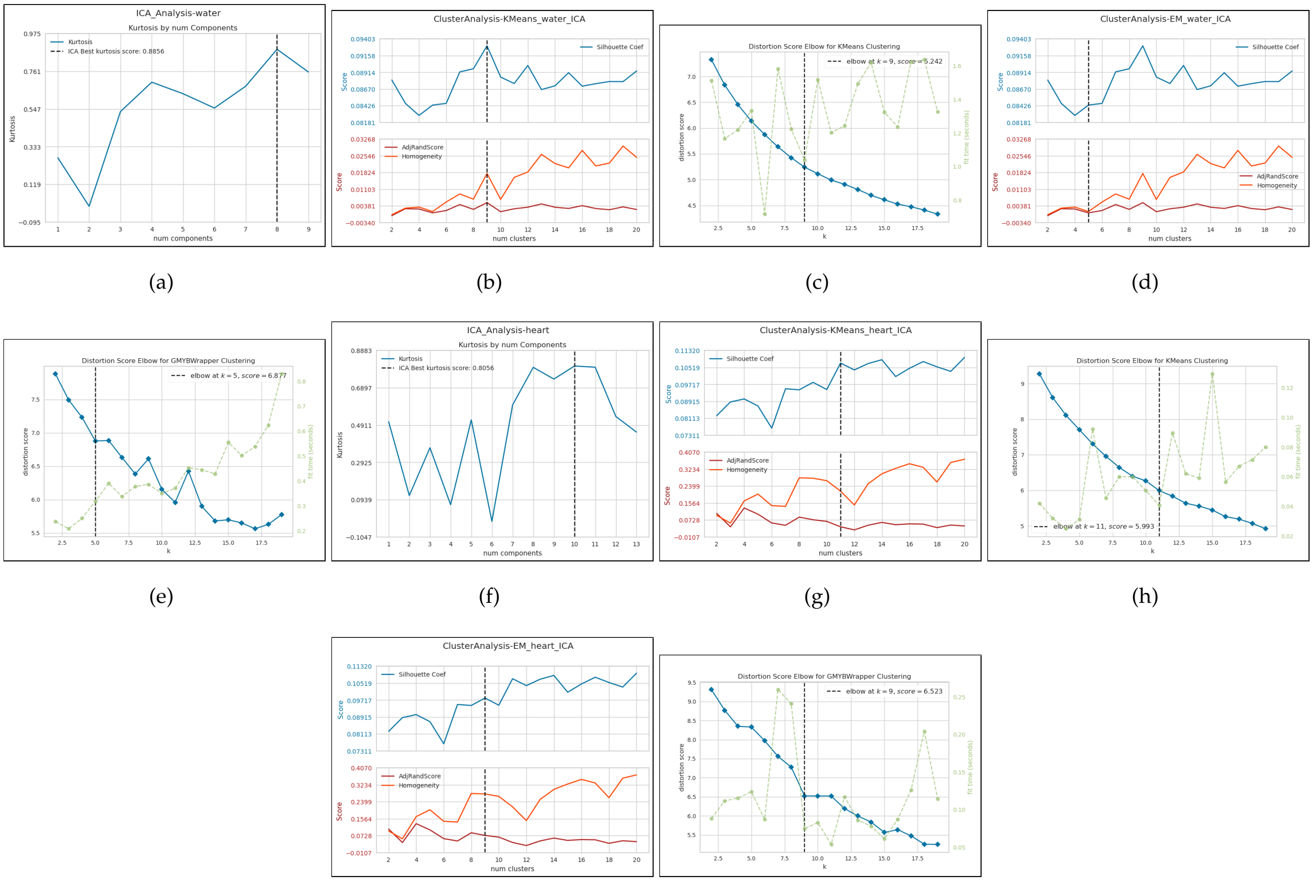


(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j)

Figure 3—PCA analysis (a) Component selection on Water Dataset (b) Metrics using K-Means on Water Dataset (c) Elbow for K-Means on Water Dataset (d) Metrics using EM on Water Dataset (e) Elbow for EM on Water Dataset (f) Component selection on Heart Dataset (g) Metrics using K-Means on Heart Dataset (h) Elbow for K-Means on Heart Dataset (i) Metrics using EM on Heart Dataset (j) Elbow for EM on Heart Dataset



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j)

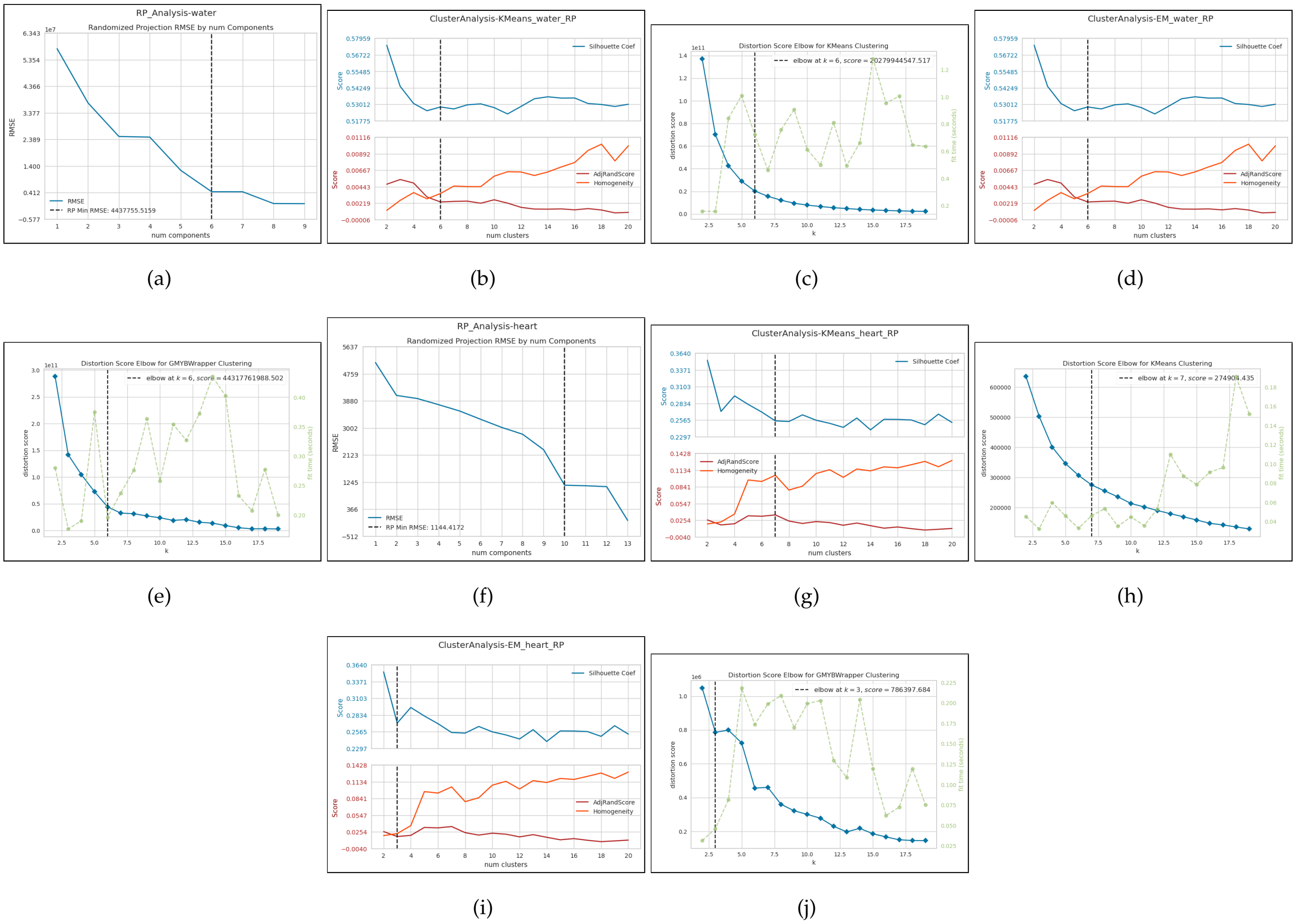
Figure 4—ICA analysis (a) Component selection on Water Dataset (b) Metrics using K-Means on Water Dataset (c) Elbow for K-Means on Water Dataset (d) Metrics using EM on Water Dataset (e) Elbow for EM on Water Dataset (f) Component selection on Heart Dataset (g) Metrics using K-Means on Heart Dataset (h) Elbow for K-Means on Heart Dataset (i) Metrics using EM on Heart Dataset (j) Elbow for EM on Heart Dataset

ICA seems to have a much more apparent affect on the datasets. For both datasets, and for both clustering methods, the optimal number of clusters is increased. However, just as before, I don't necessarily agree with some of the automated selections for optimal number of clusters. For example, the Elbow chart of EM clustering shows a very linear pattern for distortion, and the "elbow" that gets selected is just barely sharper than any other. At k=4 or 8, there are much better values for all the other metrics. Regardless, the average fit times are reduced, which is the primary goal. It is also worth noting that while the Silhouette scores are decreased, Homogeneity and Adjusted Random score are both increased after performing ICA. Since ICA isolated independence, it makes sense that resulting clusters would be more homogeneous and less random.

4.3 Randomized Projections

RP is similar to PCA because the purpose is to reduce the dimensionality while minimizing information loss, but instead of focusing on the variance, RP maintains the Euclidean distance between points when transforming to a new dimension. PCA will give theoretically better projections, but RP is a faster reduction method.

4.3.1 *Results*



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j)

Figure 5—RP analysis (a) Component selection on Water Dataset (b) Metrics using K-Means on Water Dataset (c) Elbow for K-Means on Water Dataset (d) Metrics using EM on Water Dataset (e) Elbow for EM on Water Dataset (f) Component selection on Heart Dataset (g) Metrics using K-Means on Heart Dataset (h) Elbow for K-Means on Heart Dataset (i) Metrics using EM on Heart Dataset (j) Elbow for EM on Heart Dataset

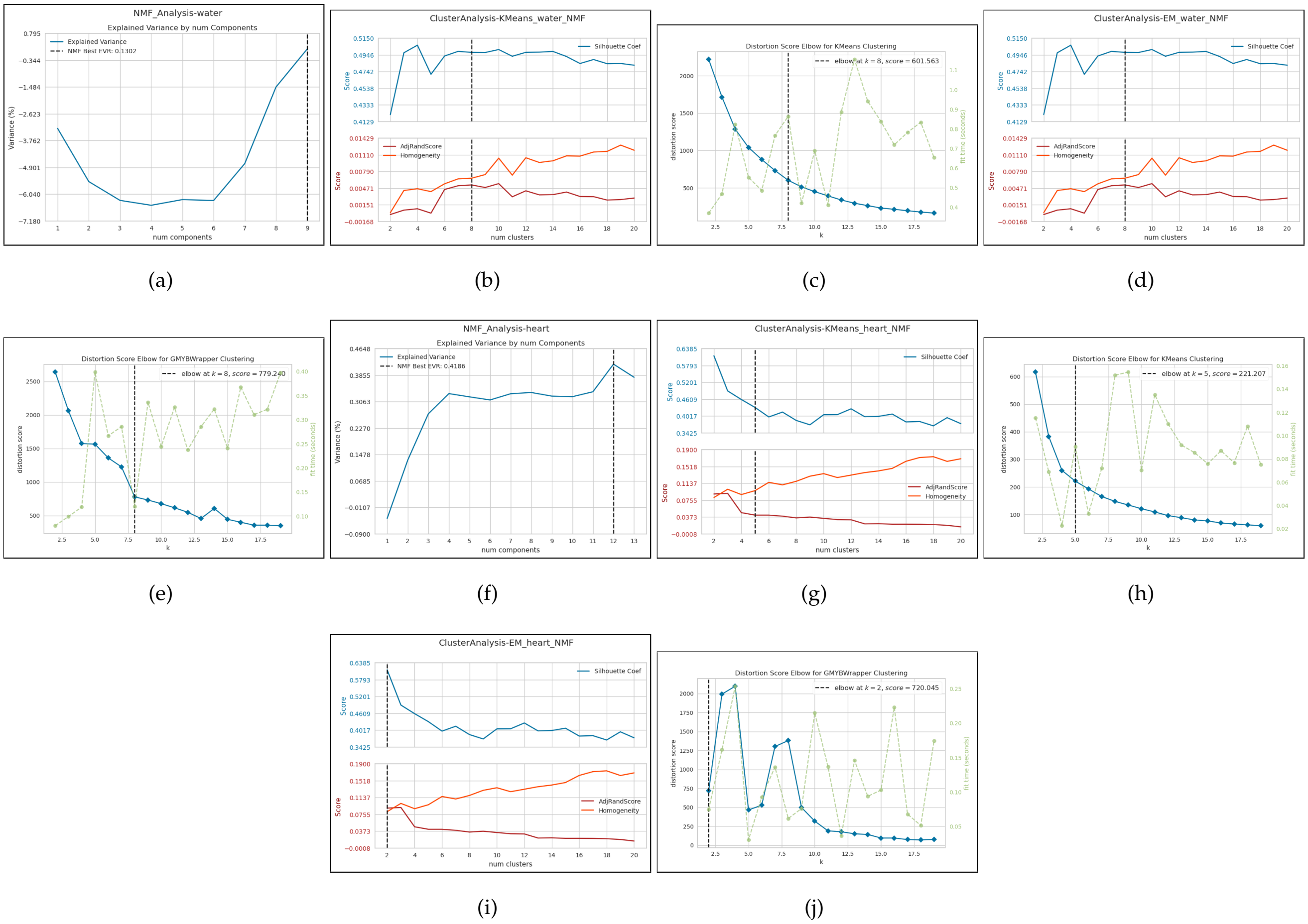
The optimal value for RP is selected by minimizing the RMSE (Root Mean Squared Error). Obviously, this is at its lowest when all components are used, so the point selected is similar to the elbow method for cluster optimization. The optimal number of components is when there is a sharp change in RMSE such that increasing the number of components has a relatively small decrease in RMSE.

For both datasets, the optimal number of components is relatively high. This means that the dataset should have minimal difference from the base data before any reductions were made. This holds true for the water dataset, as all the plots are very similar to the base version, but it is less true for the heart data. Since RP uses Euclidean distance as the measurement for information loss, and the Heart dataset has several features containing categorical data, distance is sort of imaginary. This is one of the factors I mentioned previously as why I selected the metrics that I did, as well as the datasets I chose. This method demonstrates fairly clearly that RP is less effective when Euclidean distance is arbitrary, such as with categorical data. The Water dataset is reconstructed very well in that each plot is nearly identical to the benchmark, but the heart dataset is changed significantly.

4.4 Non-Negative Matrix Factorization

NMF is a method where the input dataset is factored into multiple matrices with the property that none of the matrices have any negative elements, making them all easier to inspect. The application here is less straightforward than the previous methods, but the primary purpose is to make the data simpler.

4.4.1 *Results*



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j)

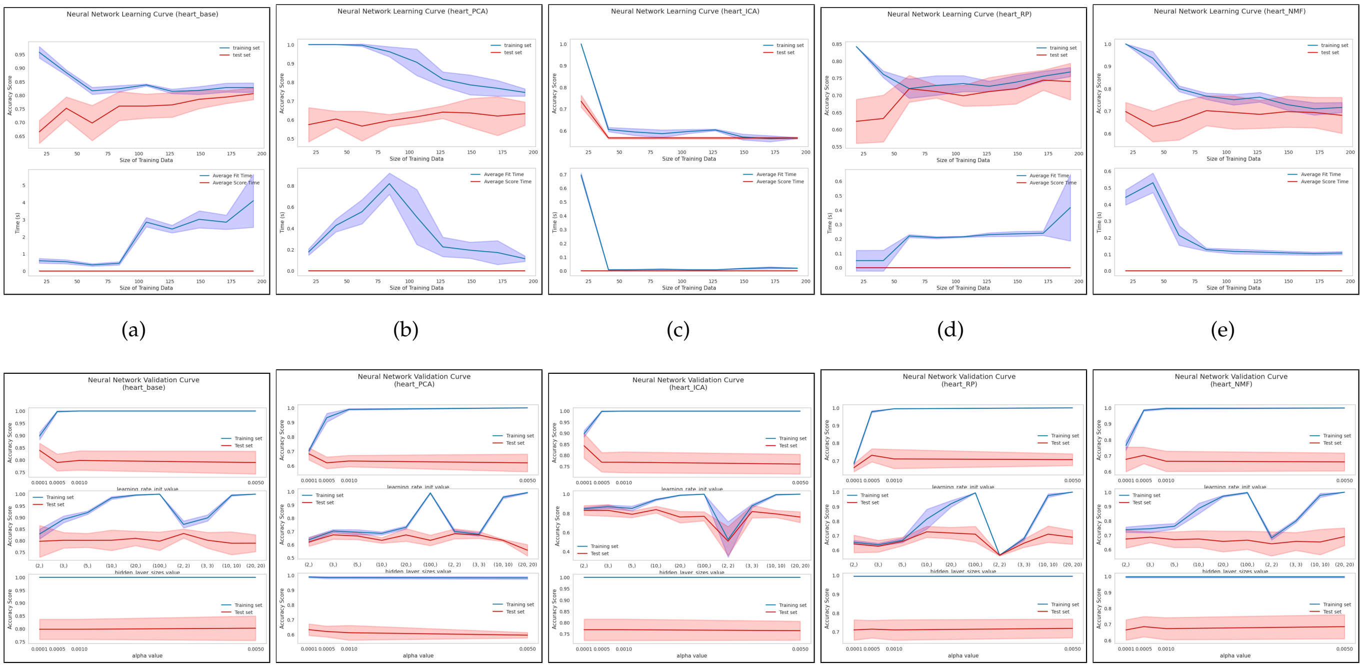
Figure 6—NMF analysis (a) Component selection on Water Dataset (b) Metrics using K-Means on Water Dataset (c) Elbow for K-Means on Water Dataset (d) Metrics using EM on Water Dataset (e) Elbow for EM on Water Dataset (f) Component selection on Heart Dataset (g) Metrics using K-Means on Heart Dataset (h) Elbow for K-Means on Heart Dataset (i) Metrics using EM on Heart Dataset (j) Elbow for EM

Optimal number of components was selected by maximizing the variance just as with PCA, however the results are not as straightforward. In fact, for the water dataset, all components were selected, but the factorization still affected the resulting model. While both datasets and both clustering methods had improvements in speed, the metrics were much more difficult to interpret, and therefore did not have very obvious "optimal" selections. As a result, while it may have done its job in simplifying the data, I would argue that, at least for this application, it did not do a great job at actually improving the model.

5 DIMENSIONALITY REDUCTION ON NEURAL NETWORKS

These same Dimensionality Reduction methods can be applied to optimize neural networks. Ideally, dimensionality reduction would improve the training time (and possibly the prediction time) with minimal impact to the accuracy. To test this out, I'm using the same Neural Network model I made for P1, and many of the same plots. Since there was some kind of issue with the Neural Networks on the Water dataset that I could not completely identify at the time, I'm using the Heart dataset to show how dimensionality reduction affects neural networks.

5.1 Results



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 7—Neural network analysis on Heart Dataset (a) Learning Curve on unedited dataset (b) Learning Curve using PCA (c) Learning Curve using ICA (d) Learning Curve using RP (e) Learning Curve using NMF (f) Validation Curve on unedited dataset (g) Validation Curve using PCA (h) Validation Curve using ICA (i) Validation Curve using RP (j) Validation Curve using NMF

Dataset	Best CV Score	Test Accuracy (%)	Train Time (s)	Predict Time (s)	Hidden Layer
Benchmark	0.8390	85.25	1.061	9.641e-4	(100)
PCA	0.6902	73.77	0.1668	6.921e-4	(10)
ICA	0.8514	63.93	0.5659	8.456e-4	(10)
RP	0.8143	75.41	0.7523	6.874e-4	(3, 3)
NMF	0.7352	50.82	0.2835	7.655e-4	(2, 2)

Although not discussed much in the previous sections, a major motivation for dimensionality reduction is to make the datasets simpler in order to allow them to work more quickly. However, this comes with a loss of information, and therefore a loss in accuracy. The goal is to maximize the runtime improvement while minimizing the hit to accuracy. Since it is difficult to measure accuracy in unsupervised learning, it can be much more clearly and concretely discussed and analyzed using neural networks.

In order to compare different methods, the table below shows the cross validation score, test accuracy, train time and predict time for each reduced model compared to the benchmark, which used unedited data and a standard NN model. It should be noted that not all of the neural networks have the same configuration. Each of them went through the same process to search through configuration parameters to maximize the CV score. I had initially thought to compare networks using the same configuration, but a key part of the process of dimensionality reduction is a simplification of the data, which means that there is likely going to be a different, simpler configuration which will maximize accuracy. Because of this, each method was optimized independently and metrics compared using the optimal configuration.

Not surprisingly, the benchmark had the most complex network, which is likely a major factor in the training and prediction times. Both PCA and ICA ended up with just a single layer of 10 nodes, which is much simpler than the benchmark. The NMF has the smallest number of total nodes, but since it has multiple layers, it's nearly the same complexity as ICA/PCA, but still much more simple than the benchmark. RP has 2 layers of 3, which is the most complex of the reduced models.

As expected, the benchmark has the highest accuracy, with a CV score of 0.84 and test accuracy of 85%. Just as with the analysis in the previous sections, NMF did not perform very well, but actually did an excellent job at reducing the training time. I'm unsure if the poor performance is due to the method itself, or if there is more tuning that I would need to do to make it work better for this situation. ICA managed to achieve a higher CV score than the benchmark, which makes sense considering that it is focused on creating well defined features through independence, but that did not translate into the testing accuracy. Both PCA and RP perform well on CV score and test accuracy, however PCA wins in performance with nearly a full order of magnitude improvement in training time over the benchmark, with only a 10% reduction in accuracy.

Although most of the results make logical sense and match my predictions, the one surprising find is how well RP performed for the Heart dataset. As previously mentioned, RP maintains Euclidean distance when reducing dimensions, but since many of the features in the Heart dataset are categorical, distance is not significant. I would have expected RP to perform quickly, but not as accurately as either PCA or ICA, but instead found that it is the slowest of the reduced models (although this is likely more related to the more complex networ configuration), and has the second highest CV score, and highest test accuracy. It's possible that there is an element of randomness which is coming into play here since both RP and Neural Networks are affected by randomness and chance, but I would still expect there to be a noticeable difference between RP and PCA, with PCA being more accurate. However, the results don't lie, and my prediction was incorrect.

REFERENCES

[1] Mitchell, Tom M. (2013). *Machine Learning*. McGraw-Hill.