Isaac Wong
Machine Learning
Assignment 3 – Unsupervised Learning

Introduction

Unlike supervised learning, unsupervised learning involves data that does not include labels, and so does not seek to approximate a function that maps data to their labels. It seeks to describe the data, and commonly includes methods for clustering and dimensionality reduction. Clustering seeks to group like data points together, while dimensionality reduction seeks to describe the data in a more compact form. We will use the same datasets from the previous paper on supervised learning, and explore the use of unsupervised learning (clustering and dimensionality reduction) on those datasets.

Datasets

Wine Quality (red): This dataset contains 1599 instances of red wine reviews and its physiochemical composition [1]. The red wines sampled were all variants of the Portuguese "Vinho Verde" wine. The dataset is interesting because there are many facets of the wine to consider when reviewing it, including its grape variety, where it was grown, how long it had been aged, and so on. The features given here are not subjective, which can be more easily controlled for during the manufacturing process. There are 11 features in the dataset, including acidity, amount of chlorides, amount of sulphates, and alcohol content. All are continuous variables, and were normalized to zero mean and 1 standard deviation. The reviews were then aggregated into two classes based on the median of the labels.

Pet Adoption: This dataset contains information and outcomes about 25,000 dogs and cats at the Austin Animal Center [2]. This dataset is interesting because a good predictive model can help animal shelter workers focus their efforts on the animals that are less likely to be adopted or returned to their owner. While the wine dataset had features that were entirely continuous, the pet adoption dataset had mostly categorical data, which required different processing and may affect the applicability of the models. Features that led to an explosion in the number of features after one-hot encoding were disregarded. 3 boolean variables and 1 continuous variable remained. The continuous feature was normalized, and the classes aggregated into 2 outcomes. The data was only slightly skewed, with about 10,000 good outcomes, and 15,000 bad outcomes.

Methods

The algorithms were implemented in Python, with the help of the Scikit-Learn, Matplotlib, and Pandas library. We explored various clustering and dimensionality reduction methods. We then performed clustering on the feature-reduced dataset, and also trained a neural network on both the feature-reduced dataset and the clusters produced after dimensionality reduction.

Clustering

Clustering algorithms group data points in such a way that data points in the same group are more similar to each other than data points in other groups. Since the data does not have labels, what constitutes a cluster is ambiguous and different algorithms have different notions of what makes points similar. We explored 2 algorithms: k-means and Gaussian mixture models (GMMs). Implicitly, k-means minimizes the distortion of the data points with respect to their clusters, and is a "hard" clustering algorithm (assigning points to clusters with no ambiguity). GMMs aim to fit a number of Gaussian distributions to the data, and is computed using the expectation-maximization (EM) algorithm, which produces a maximum likelihood estimate of the model. Since it fits a mixture of probability distributions, GMMs are a "soft" clustering algorithm.

We used both external and internal metrics for analysis – computed with and not with respect to the true labels, respectively. The silhouette score compares the mean inter-cluster distance to the mean nearest-cluster

distance for each point. Distortion is relevant for evaluating k-means. Akaike and Bayesian Information Criterion (AIC, BIC) are ways to score the quality of statistical models given a dataset, and is relevant for GMM. The AIC and BIC are similar, but BIC has a larger penalization term when the number of data samples is large [3]. We aim for a high silhouette, low distortion, low AIC, and low BIC score.

Homogeneity measures how pure the clusters are. Homogeneity tends to increase with number of clusters, since n clusters for n data points is perfectly homogeneous. Completeness measures how spread out the data points for a single label are among the clusters. This has the opposite behavior from homogeneity, where having a single cluster means that all labels are perfectly complete. The adjusted rand index (ARI) is a measure of similarity between 2 clusterings, adjusted for randomness. In all 3 metrics, the higher the score, the better.

We clustered both datasets with k-means, with k ranging from 2 to 50. The results of the internal metrics are shown in figure 1. Euclidean distance was used to compute the silhouette score, but it's not a great measure of distance when it comes to Boolean variables. Although the pet dataset is composed of mostly Boolean variables, Euclidean distance was still used because of the presence of some continuous variables and the fact that k-means inherently uses Euclidean distance.

We can see that for the wine dataset, the distortion curve has a less obvious elbow point compared to the pet dataset. The elbow is the point where adding an additional cluster only marginally improves the distortion score. Since we tend to think that simpler models are better, the elbow lets us choose a number of k that gives us a good distortion score while minimizing k. This difference in elbow shape may indicate that the data points are not distributed into clear clusters for the wine dataset, while the pet dataset is more easily clustered. However, we can still approximate a "good" k for both datasets: 9 for wine and 10 for pet.

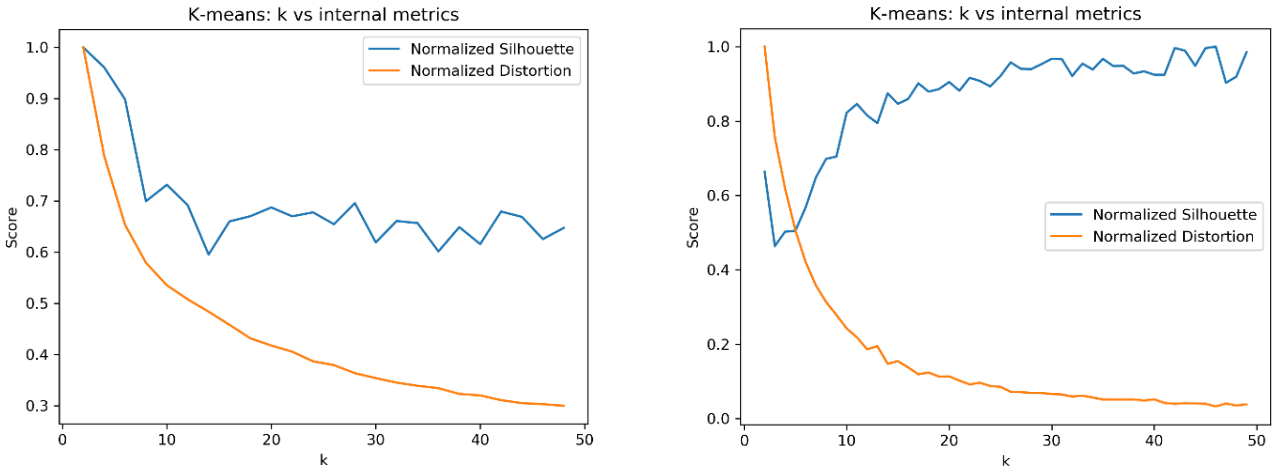

Figure 1: Silhouette and distortion scores for k-means with various number of clusters (k) on the wine (left) and the pet (right) dataset

The values of the silhouette score also indicate which dataset is more easily clustered. The maximum silhouette score for the wine dataset is 0.214, while the maximum score for the pet dataset is 0.753. The score is 0.152 at k=9 for the wine dataset, while the score is 0.619 at k=10 for the pet dataset. The higher score means that the points within clusters are more similar to the points between clusters for the pet dataset. Looking at the curves for the silhouette score, the pet curve takes a more expected trajectory – increasing as k increases and distortion decreases, and then hits a plateau. The wine curve is more unusual, starting high and then dropping to a plateau. This may be because of the difficulty in clustering the dataset.

The external metrics are shown in figure 2. We see immediately that, as explained above, homogeneity scores increase as k increases. We do see somewhat of an elbow shape, occurring close to the peaks in the ARI and completeness scores. This elbow point is much more pronounced in the pet dataset, similar to the distortion curves.

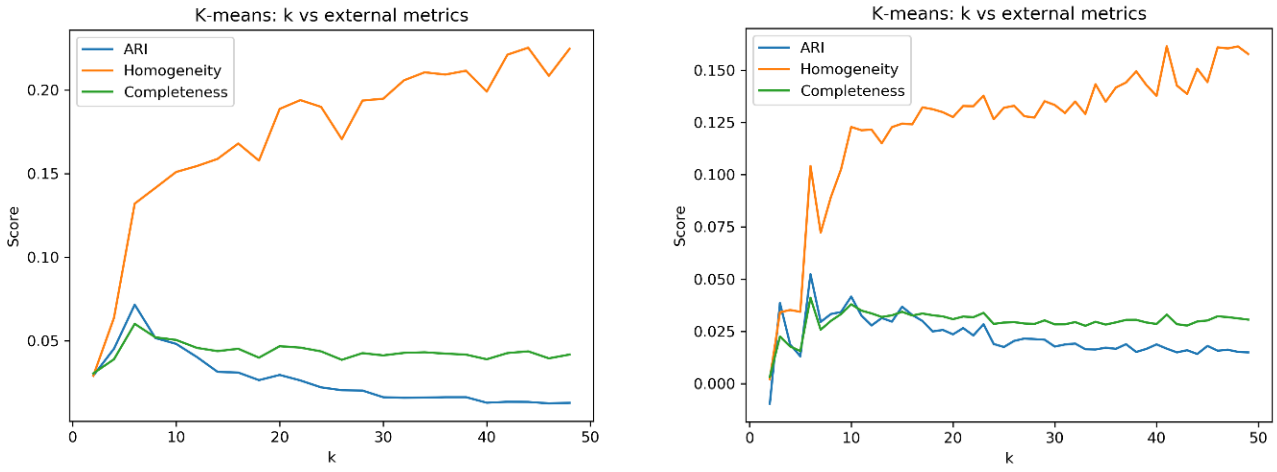

Figure 2: ARI, homogeneity, and completeness scores for k-means with various number of clusters (k) on the wine (left) and the pet (right) dataset

We also see that the scores are very low in general, even for the pet dataset, where we had a fairly high silhouette score. This is because what makes for a good cluster based on how internally similar they are may not translate to actual labels. If the features are not effective at predicting the labels, then even if they can be easily clustered, it does not mean that the clusters correspond to the labels. Nonetheless, for both datasets, we see a maximum at the same place for the completeness and ARI metrics. The maximum ARI score is 0.0757 at k=5 for the wine dataset, and 0.0524 at k=6 for the pet dataset. There is a difference between the best k value found by the internal and external metrics, but since we are trying to find the clusters that best fit our labels, then the k found by the external metrics is more appropriate.

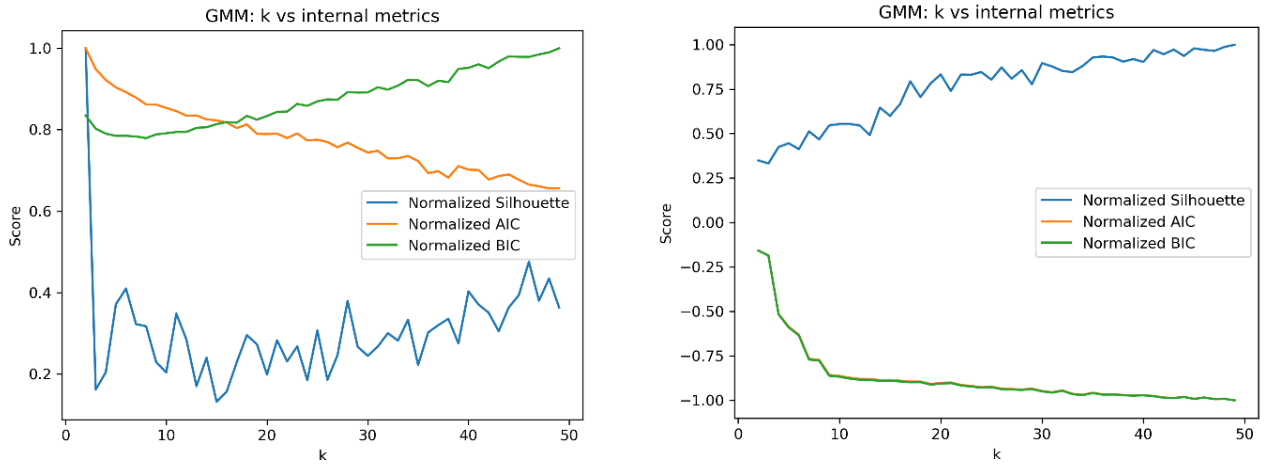

Figure 3: Silhouette, AIC, and BIC scores for GMM with various number of clusters (k) on the wine (left) and the pet (right) dataset

We also clustered the datasets with GMM, estimated using EM, with k ranging from 2 to 50. The results of the internal metrics are shown in figure 2. We see that for the wine dataset, the AIC and BIC curves are different, with the AIC steadily decreasing with a weak elbow point and BIC having a minimum at k=8. The penalty for having more parameters is greater in BIC compared to AIC, which explains the upward trend as k increases. Like for k-means, the value of the silhouette score is very low, with a maximum of 0.235 and a score of 0.075 when k=8. The weak elbow of the AIC and the low silhouette score indicates that the algorithm is not able to cluster the data points well. For the pet dataset, we see that the AIC and BIC curves are almost identical, with a fairly sharp elbow at k=7. The silhouette score is slightly higher, with a maximum of 0.723 and a score of 0.370 at k=7.

The external metrics are shown in figure 4. Just like before, we see that the homogeneity scores steadily increase, with a much sharper elbow for the pet dataset. The elbows occur close to the peaks for the ARI and completeness scores. The value of the scores are higher than before, but still fairly low. This indicates that the clusters do not correspond well to the labels. Nonetheless, the best k can be estimated at the peaks for the ARI score. The ARI score is 0.126 at k=3 for the wine dataset, and 0.0843 at k=6 for the pet dataset. The k's found using the external and internal metrics is similar for the pet dataset, but not for the wine dataset. Since we are trying to find clusters that best fit our labels, the k found by the external metrics is more appropriate.

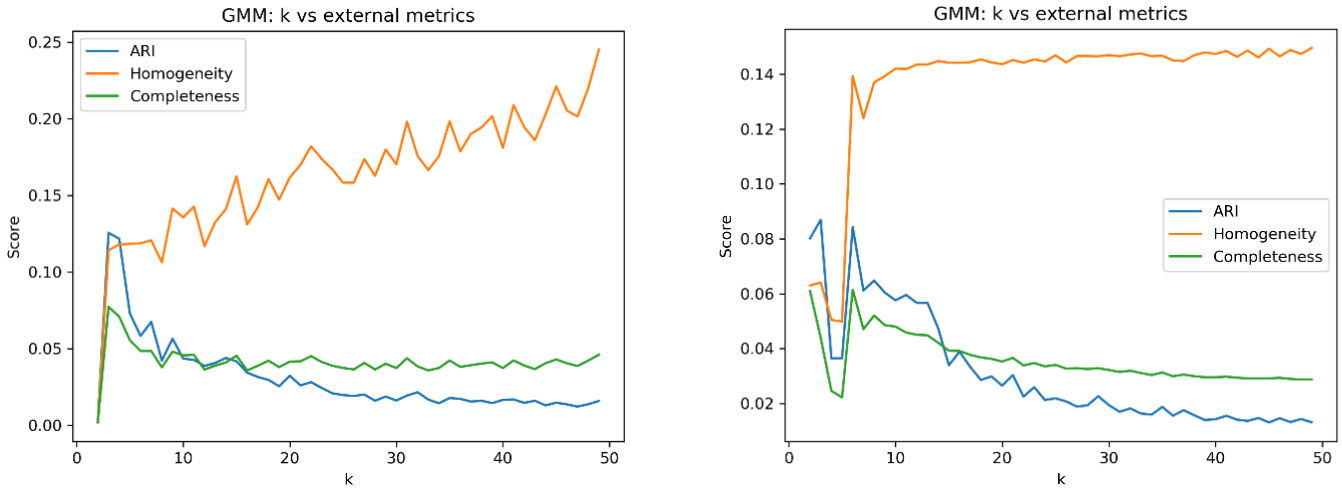

Figure 4: ARI, homogeneity, and completeness scores for GMM with various number of clusters (k) on the wine (left) and the pet (right) dataset

Dimensionality Reduction

Dimension reduction is a way of describing the data in a more concise manner. It aims to reduce the number of features needed to describe the data, either through filtering (removing features), or transformation (combining features into fewer number of features). We explored 4 dimensionality reduction algorithms: principal component analysis (PCA), independent component analysis (ICA), random projections (RP), and random forest (RF). The first 3 are feature transformation techniques, while the last is a feature filtering technique. We evaluated the algorithms through their individually appropriate metric (variance explained for PCA, etc.), and also by training a boosted decision tree algorithm. The boosting model was trained across various number of components output from the dimensionality reduction algorithm.

PCA is a feature transformation algorithm that takes the original features and transforms them into a set of features that are linear combinations of the original features, in such a way that the principal feature explains the largest possible variance. The 2nd principal feature explains the next largest possible variance, and so on. For PCA to work well, the data should be normally distributed. Figure 5 (top) shows the data after transforming, using the top 2 principal components and their true labels. We can see that the data is very non-clustered over the 2 components, though it could become more clustered with more components. The strange pattern seen in the pet dataset is due to the Boolean variables in the data. Figure 5 (bottom) shows the result of running PCA on both datasets, and then using the top *m* components to train the boosting model. Accuracy score was used as the metric for evaluating the model.

We can see that for the wine dataset, most of the variance is explained with 3 principal components. We can also see that effect on the accuracy of the boosting model, since the accuracy increases up until the 3rd principal component is added, and then plateaus as more are added. On the other hand, most of the variance in the pet dataset is explained with just 1 principal component. The accuracy of the boosting model is also high with just one component, and then barely increases as more are added, and plateaus at 3 components. Since we would like to reduce the size of our dataset but still keep a high accuracy, we can reduce both datasets to 3 components using PCA.

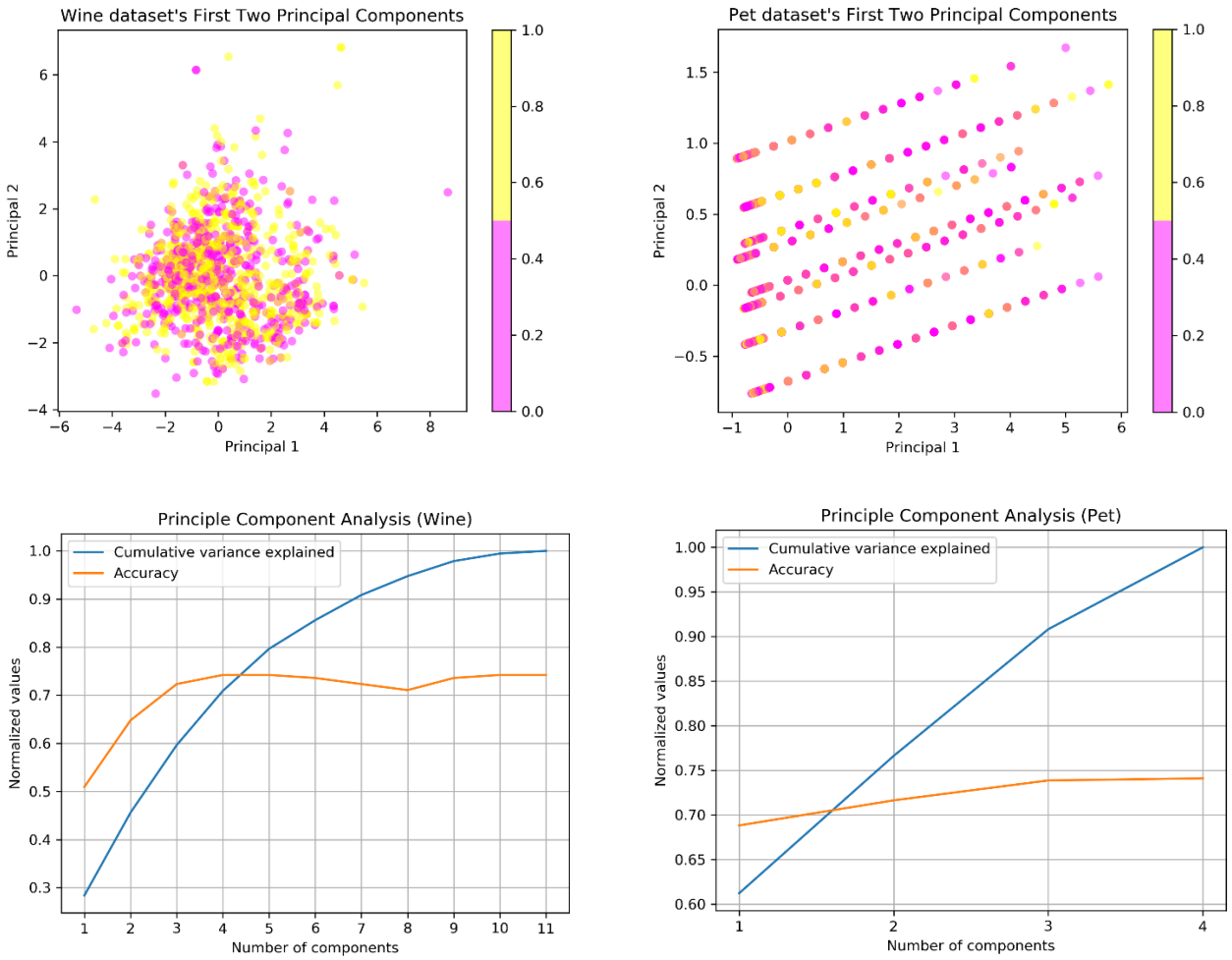

Figure 5: Principal component analysis on the wine (left) and the pet (right) dataset. Accuracy is computed from a trained boosted decision tree.

ICA is a feature transformation algorithm that takes the original features and tries to find the independent components that linearly combine to contribute to the original features. These independent components are assumed to be non-Gaussian. Contrary to PCA, where the output components tend to be Gaussian, the output components for ICA tries to be as non-Gaussian as possible. We therefore measured the mean excess kurtosis of the data after being transformed by ICA, as a way to measure non-Gaussianity. Figure 6 shows the result of running ICA on both datasets with different output number of components, and then training the boosting model.

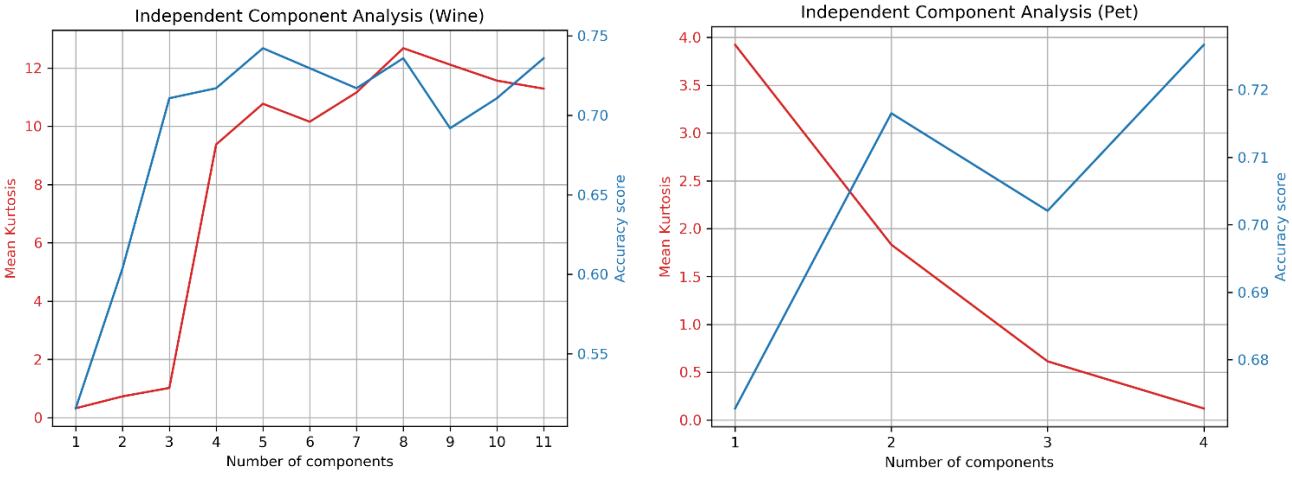

Figure 6: Independent component analysis using different number of output components on the wine (left) and the pet (right) dataset. Kurtosis is computed from the transformed training data, then averaged over components. Accuracy is computed from a trained boosted decision tree.

We can see that for the wine dataset, average kurtosis jumps dramatically at 3 components, and then slowly increases as the number of components increase. The accuracy score also increases at 2 components, and then plateaus. Although the basis of ICA is quite different from PCA, the number of components that work well to describe the data is the same for both techniques. We also find a similar result for the pet dataset. Kurtosis is highest for 1 component, which is the same result given by PCA. However, we see that the accuracy score increases as the number of components increase, though the average kurtosis decreases. The largest increase in accuracy appears to be with 2 components. There is another increase in accuracy at 4 components, but the value of the increase is small and keeping 4 components would mean no reduction at all. For the wine dataset, ICA appears to work fine with 3 components.

RP is a feature transformation algorithm that takes the original features, and projects them onto randomly generated vectors. The benefit of RP is the speed compared to ICA and PCA, which can take some time to compute. It also happens to work well, by reducing the number of dimensions while also managing to capture some of the information in the original features. There is no obvious metric to measure the goodness of the chosen projections, since they are randomly selected, so we evaluated the algorithm by just computing accuracy using the boosting model. Figure 7 shows the result of running RP on both datasets, using a different number of projections, and then training the boosting model. RP for each number of projections was run 10 times to reduce the variance of the accuracy.

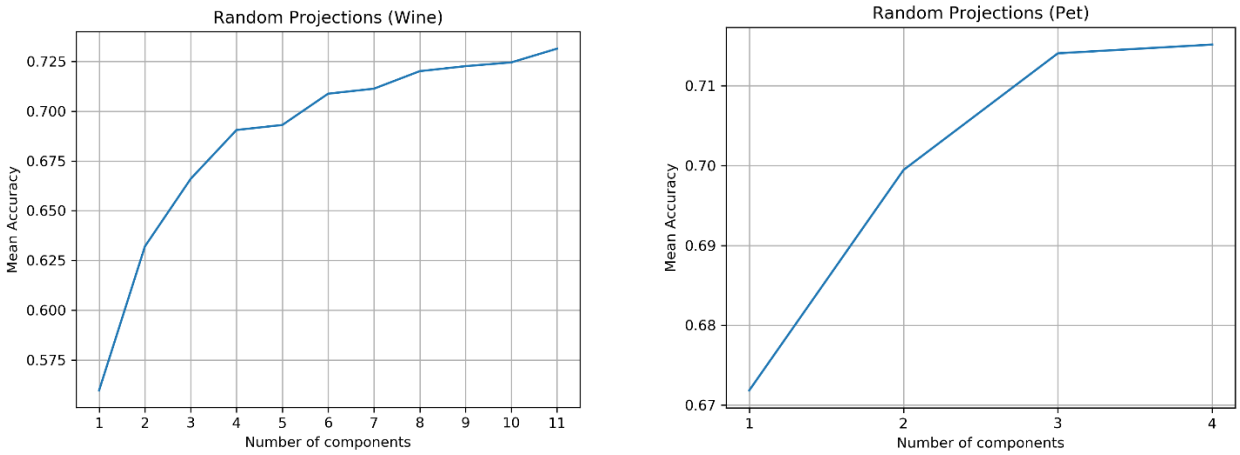

Figure 7: Random Projections using different number of output components on the wine (left) and the pet (right) dataset. Mean accuracy is computed from boosted decision trees trained over 10 iterations of RP per number of components.

We can see that for both datasets, there appears to be an elbow curve of accuracy. For the wine dataset, the accuracy increases dramatically from 1 to 4 components, and then the improvement starts to taper off. For the pet dataset, the accuracy increases dramatically from 1 to 3 components, and then plateaus. For the wine dataset, a good number of components would be 4, which is only slightly higher than the number of components obtained from PCA and ICA (3). We should expect a higher number for RP, since the selection was just random, but the performance is surprisingly good. The value of the accuracy is also approximately the same as the PCA and ICA. For the pet dataset, we can see that the shape of the accuracy curve is similar to the PCA accuracy curve. At 3 components for both PCA and RP, the accuracy plateaus. Just like with the wine dataset, the performance and accuracy of RP is similar to a more complex algorithm like PCA.

Random forests are usually used as a classification model, but we can use it as a filtering algorithm, since implicit to the model is feature selection for each tree. After training, we can then extract the most important features computed by the random forest using the GINI index. One benefit to using filtering techniques is that the resulting components are more interpretable. One disadvantage is that filtering techniques tend to be greedy, and we may easily fall into local optima. Figure 8 shows the cumulative importance given to each of the features, and the accuracy of the boosting model trained on the top *m* number of components.

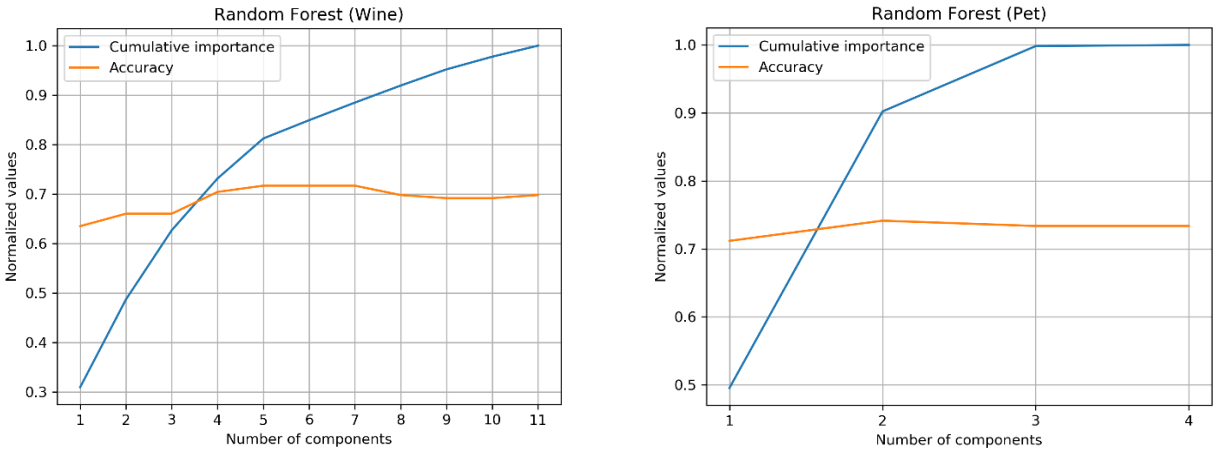

Figure 8: Random Forest on the wine (left) and the pet (right) dataset. Accuracy is computed from a trained boosted decision tree.

We can see that for both datasets, the cumulative importance curve appears to have an elbow point. The elbow is much more pronounced in the pet dataset, with an elbow at around 2 components. The wine dataset also has a gentle elbow at around 5 components. We also see that the accuracy increases slightly with those number of components. In addition, the accuracy values are comparable to the previous feature transformation techniques. The reduced number of components in greater than the other techniques for the wine dataset, perhaps because some of the important features could be combined. The reduced number of components is comparable to the other techniques for the pet dataset, but that can be expected since the original number of features is very small, and there is not much room to reduce.

In table 1 is a summary of the estimate of the best number of features for each dimension reduction algorithm. We can see that the number of features is reduced drastically for the wine dataset, and also to some extent in the pet dataset. A smaller reduction percentage may be due to the fact that the pet dataset has a small number of features to begin with, and there is not much room to reduce. Even with the reduction, in both datasets the accuracy obtained was decent, and similar across the board.

Table 1: Estimated best number of features for each algorithm					
Number of features for:	*Original*	*PCA*	*ICA*	*RP*	*RF*
Wine dataset	11	3	4	5	5
Pet dataset	4	3	2	3	3

Clustering and Dimension Reduction

We would like to know whether dimensionality reduction improves clustering, so we ran dimension reduction using the 4 techniques and using the parameters for number of components determined in the section above. We then ran 2 clustering algorithms: k-means and GMM using EM. They were evaluated using just the ARI for an external metric, since that seemed to be the easiest to interpret. K-means was evaluated with distortion, and GMM with BIC.

The scores for k-means are shown in figure 9. We can see that all the reduced datasets have an elbow curve that is more pronounced than the original dataset, which indicates that the data is more easily clustered. However, that does not necessarily mean that the clusters are good with respect to the labels. In the ARI curves for the wine dataset, we can see that the best scoring algorithm is RF with a peak at k=3, and the distortion curve for RF is one of the gentlest. We also see that the ARI scores for RP is terrible across all values of k. A high scoring RF and a low scoring RP may be indicative that there are several useful features in the dataset, and several useless features. By selecting for useful features through the RF, we eliminate the useless ones from consideration and improve ARI. By

randomly projecting through the space, we also project over the useless dimensions, diluting the effectiveness of the useful features.

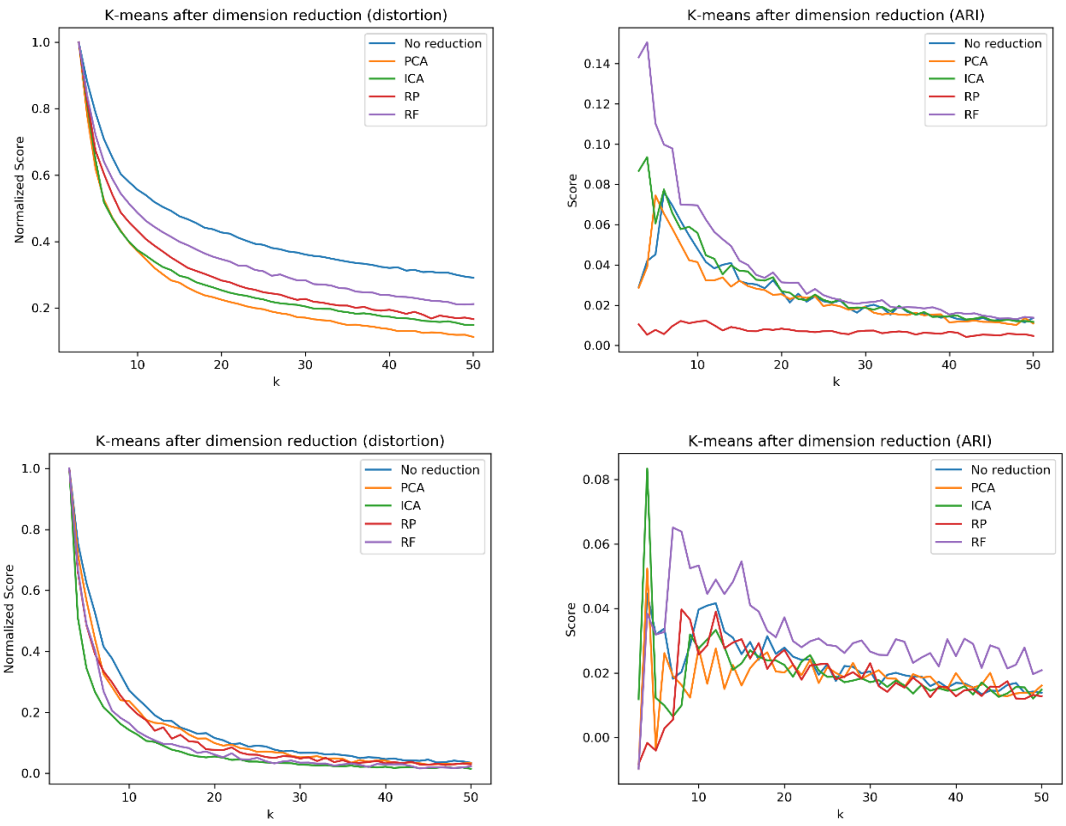

Figure 9: Distortion(left) and ARI(right) metrics for the wine(top) and pet(bottom) dataset.

For the pet dataset, we see that the distortion curves are very similar, and so are the peak ARI values. There appears to be a peak in ICA and PCA at k=3, and a secondary peak at the same k for the original data and RF. RP has a peak at a larger value of k. Based on the similarity of the distortion and ARI scores, it appears that any algorithm would work reasonably for k-means clustering over the pet dataset.

The scores for GMM are shown in figure 10. We can see that the AIC values after reduction on the wine dataset is less than the AIC on the original dataset. This indicates that for the wine dataset, there are some features that are better for clustering and some that are not. Reducing the dimension and throwing away some features makes clustering more effective. RF results in the highest peak in the ARI curves, and follows the same reasoning as the previous section. For the wine dataset, it looks like there is a slight elbow at the AIC curves that correspond to the peaks in the ARI curves at k=3.

For the pet dataset, the AIC values after reduction is actually higher than the AIC values on the original dataset. This may indicate that throwing away some of the data in the reduction makes the data less clustered. However, we do see elbow points at small k, which is also where most of the peaks in ARI happen. There does not appear to be a clear relationship between good AIC and good ARI scores, which is unsurprising since one is calculated with respect to the labels and the other is not. The best ARI score is from RP at k=5 and RF at k=4. The high score of RP is surprising when it did not do well in k-means clustering, but this high performance may be due to the random projections into good directions. The curve is very similar to the RF curve, so it may have happened to project into similar directions as the RF selected components.

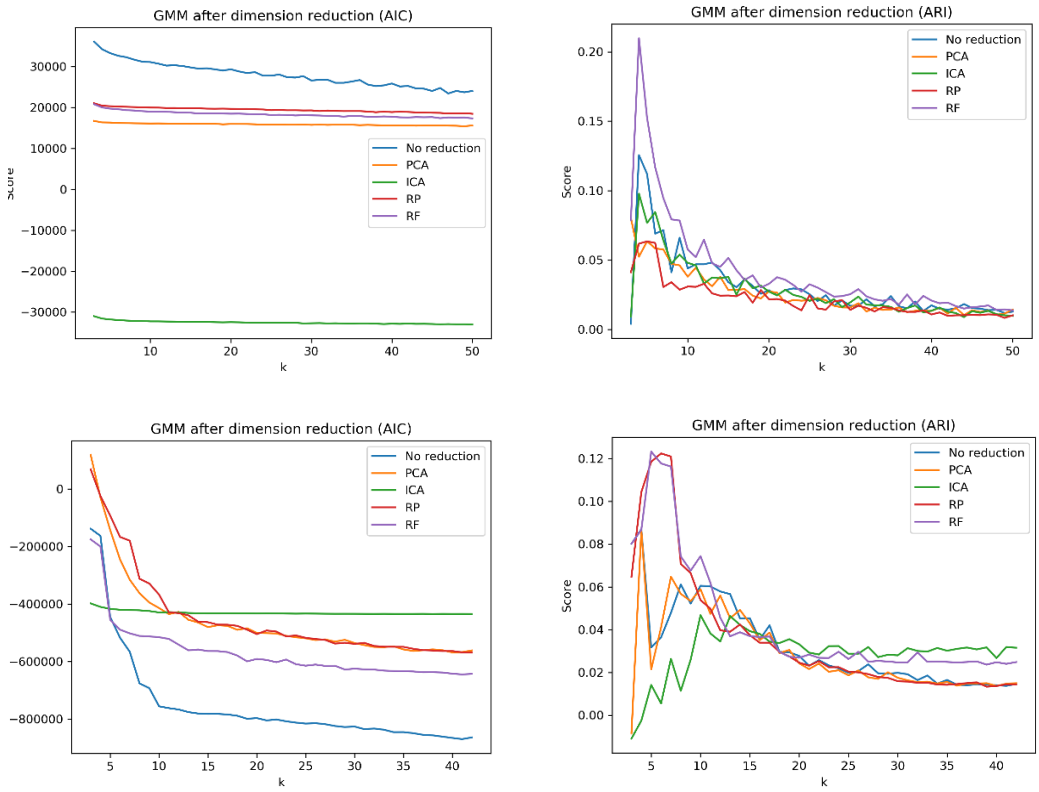

Figure 10: AIC(left) and ARI(right) metrics for the wine(top) and pet(bottom) dataset.

Table 2 summarizes the best k for each clustering algorithm as determined by the peak in the ARI score. Overall, there is not a great difference between the best k for the original dataset and the reduced datasets. We see a slight increase in number of clusters for RP, and a slight decrease in number of clusters for PCA and ICA.

Table 2: Best k determined from ARI score peaks for the various algorithms					
Best k (ARI) for:	*Original*	*PCA*	*ICA*	*RP*	*RF*
(Wine) k-means	5	4	3	No clear peak	3
(Wine) GMM	3	2	3	4	3
(Pet) k-means	3	3	3	7	6
(Pet) GMM	3	3	9	5	4

Neural Net Training

The neural network that performed well on the wine dataset was determined to be a 4 hidden layer network with 25 hidden units each. We would like to know whether dimension reduction or clustering can improve the performance of the neural network by reducing the number of features of the dataset. We tested this by doing 10-fold cross validation with the 4x25 architecture, using datasets reduced by the 4 dimensionality reduction techniques in the previous sections, and also on datasets that were clustered after being reduced. The 2 clustering algorithms above were used, and the clusters were one-hot encoded into features before training.

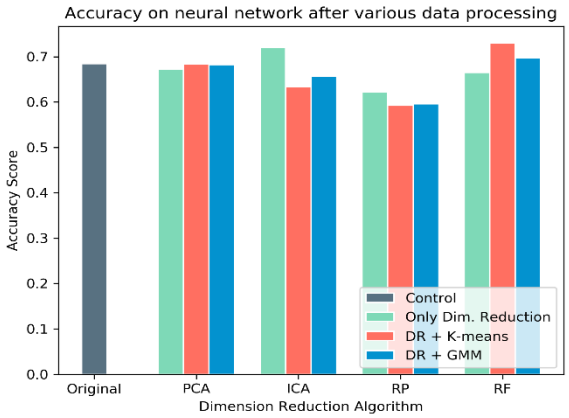

Figure 11: Accuracy on neural network after dimension reduction and clustering algorithms

We can see from figure 11 the testing accuracies obtained after various reduction/clustering combinations. The bar on the far left indicates the accuracy on the original, untouched dataset, which serves as a control. We can

see that RP performs worse and PCA performs almost identically across the board. Previously, we had suspected in the section for clustering and dimension reduction that the dataset contains features that are useful and features that are not. The bad performance of random projections may be attributed to the diluting of the useful features with the useless features, and the similar performance of PCA may be due to the algorithm capturing enough of the important features in the principal components.

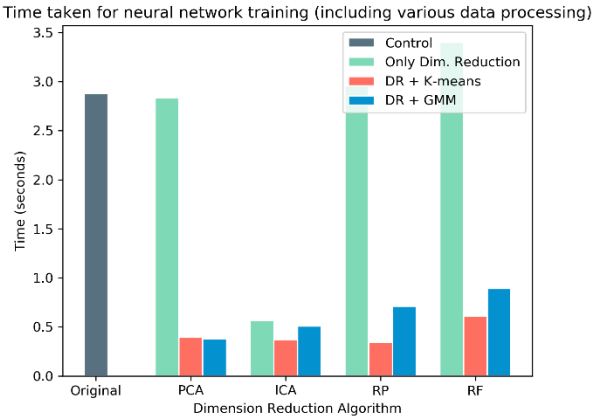

Figure 12: Time taken for neural network training, including data processing (dimensionality reduction and clustering)

ICA by itself performs well, but we can also see a decrease in accuracy after clustering. In the section for dimension reduction, we saw that the kurtosis for ICA was very high, indicating that the dataset may have a good separation into statistically independent sources. This good separation may have contributed to the improvement in accuracy, but realistically the improvement is not high. The decrease in performance after ICA may be due to the attempt to fit a Gaussian model or a similarly shaped K-Means model to the very non-Gaussian components that ICA produces. Finally, we have RF, which shows similar performance for just dimensionality reduction, but improved performance after clustering. As mentioned before, we expect that there are some features that are useful and some that are useless, and RF would tend to pick the more useful ones. We see a similar performance, perhaps due to the neural network already learning which features are useful and which are not. The increase in performance is small, but difficult to explain. We speculate that the clustering might lead to grouping the wine into various subgroups based on their physiochemical composition, and those subgroups are more predictive themselves than their physiochemical composition.

In figure 12, we have the time taken for the combination of dimension reduction, clustering, and training. Some of the times look unusual – for example, the drop in time taken from PCA only to PCA and K-Means. Even though the addition of clustering makes it seem like it should taken longer, the bulk of the time consumed by the PCA-only model is in neural network training. After clustering, the features were one-hot encoded, and it appears that the Boolean features make neural network training quicker. We can see this drop across the board for all four dimensionality reduction algorithms. The only dimensionality reduction algorithm that resulted in a much quicker training time was ICA.

Conclusion

Dimension reduction has the benefit of reducing the number of features and the curse of dimensionality, with the tradeoff that we would have to explore which algorithm would be appropriate and still give us enough information to get good accuracy. For our datasets, the accuracy was not improved too much with dimensionality reduction or clustering after dimensionality reduction. However, a benefit is in the training time. If the right dimension reduction or clustering algorithm is chosen, the training time can be drastically reduced for not too much loss in accuracy.

Sources

[1] Wine quality dataset source: https://archive.ics.uci.edu/ml/datasets/wine

[2] Pet adoption dataset source: https://www.kaggle.com/c/shelter-animal-outcomes

[3] AIC vs BIC: https://www.youtube.com/watch?v=4al2LfJz6Q8