Dimensionality Reduction

Cody Phrampus
cphrampus3@gatech.edu

1 Dataset 1 - Wine

This dataset contains 13 features describing wines mapping to 0, 1, or 2 indicating the location. The set is fairly small with only 178 data points, but the classes are covered roughly equally (Wine).

1.1 Clustering

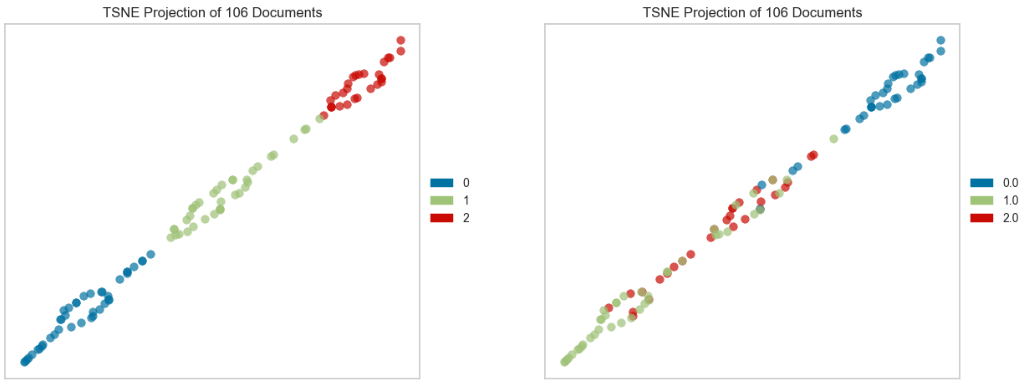

Figure 1, 2—TSNE of the clusters colored with predicted labels (left) and true labels (right)

In order to find the appropriate k for k-means, the elbow method was used on graphs of distortion and inertia over a range of ks. Two graphs were used to double check that a certain value worked well, as well as to break ties on ambiguous values on one graph. This method yielded fairly good results in terms of agreeing on a number for k as well as aligning with the number of classes in the data, 3, but the algorithm doesn’t know about those. As can be seen in the TSNE plot above, the clustering seems fairly reasonable given this mapping of the higher dimensional space, with perhaps some debate on the rightmost middle points. However, coloring the plot with the true labels shows that the data are much interspersed in a way that does not lead to “reasonable” clusters, again given this particular visualization of all the present dimensions, resulting in a score of just under 73%. Given this odd ground truth, it is possible that the dimensionality reduction algorithms will give a more “reasonable” grouping after some excess dimensions can be stripped away, as those additional dimensions could be making it difficult to accurately see natural clusters using TSNE. The smaller number of items in this dataset also allows a much easier time visually identifying potentially good clusters, at least visually appealing ones. This particular dataset also has points that clustered reasonably well, where someone could look at the uncolored points and see about 3 clusters, one in the middle and one on either side, even though this is not consistent with ground truth labels.

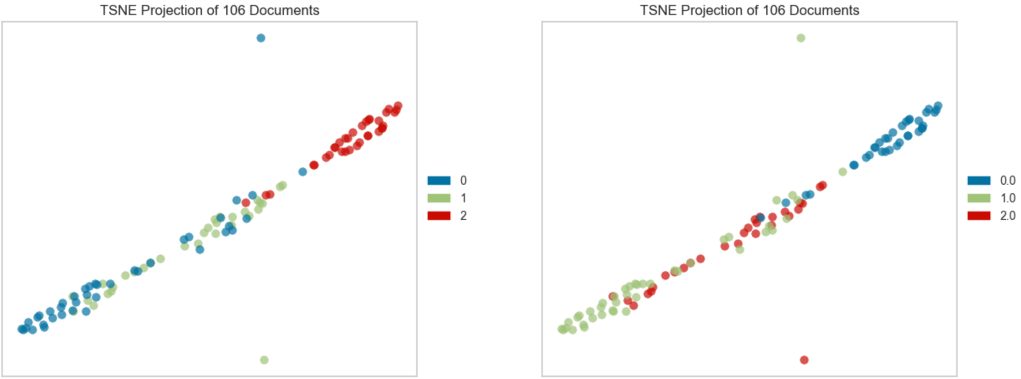

Figure 3, 4—TSNE colored with EM predicted labels (left), and True labels (right)

In order to find the best number of clusters and the best covariance type for EM, the scikit-learn example for Gaussian Mixture Model Selection was used, which plots the BIC scores for each available covariance type for each number of clusters (Gaussian Mixture Model Selection). K was chosen to be 3 along with a diagonal covariance model from the aforementioned plot, excluded for space. As with k-means, EM found a number of clusters equal to the number of classes in the data, despite not knowing about them. This seems to indicate that data occupy relatively different areas within their space, providing a relatively clear grouping. The TSNE plots are messier than the ones for k-means, but performed much better, getting a score of 95%, despite the mismatch to the true labels making the coloring different. This big difference seems to indicate that this data set is more amenable to softer clustering, perhaps due to allowing the clusters more freedom to settle into a good space, without having to fully commit. It is also worth noting that because this method is treating the points as having been generated using a gaussian, this could be better modeling the underlying natural processes and the fact that boundaries between areas of land are hazier than the clear cut lines that k means assumes.

1.2 Dimensionality Reduction

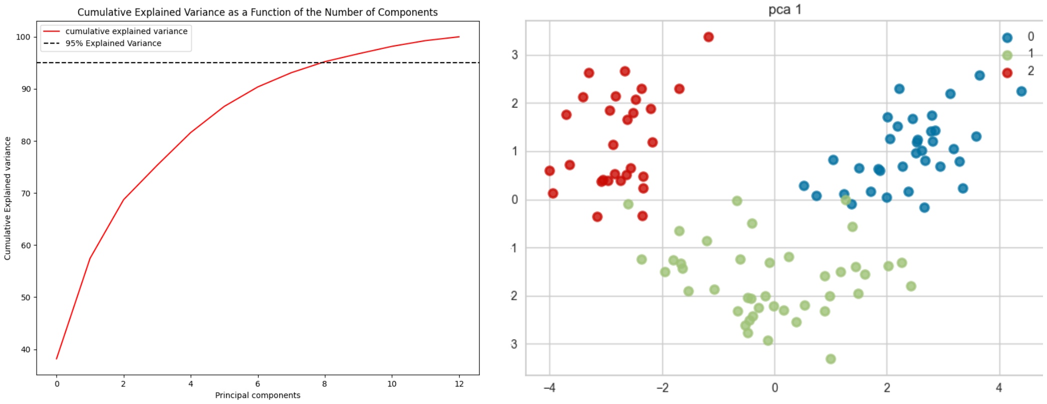

Figure 5, 6—variance over components (left) and the first 2 dimensions for PCA (right)

Because PCA uses the eigenvalues/variance to order dimensions in order to allow the least useful ones to be discarded, the cumulative variance over these dimensions was plotted in order to determine how many dimensions would be sufficient. 6 dimensions ended up being selected, with a total variance of about 90%, leaving half of them to be discarded. This method would be expected to work fairly well, if we can remove dimensions that are not changing much, thus likely not influencing the ultimate label, we can operate in a more reasonable space. As can be seen from the plot above, the first two dimensions of the transformed data provide fairly good separation, aside from a few green points, but these may be separated from the red and blue points in the third or fourth dimensions of the data. This resulting projection also results in a reconstruction error of about 12%.

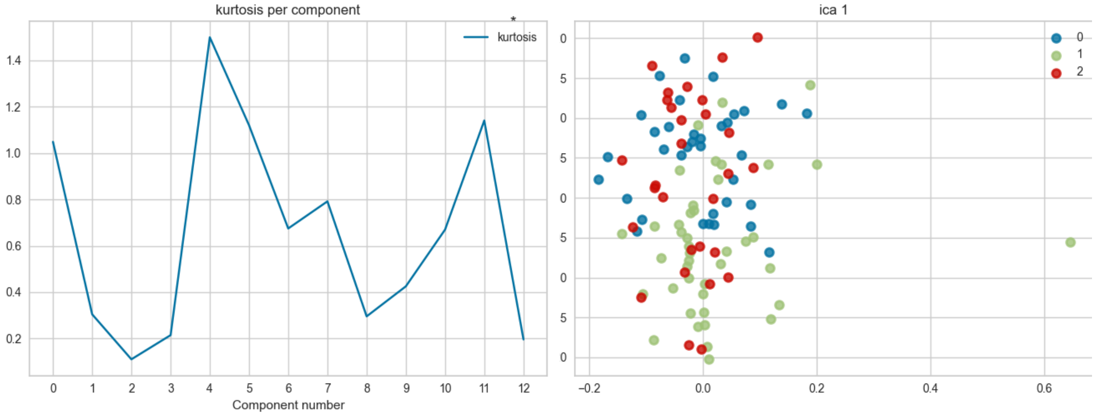

Figure 7, 8—kurtosis per component (left) and the first 2 dimensions for ICA (right)

ICA, as opposed to the variance based methods of PCA, attempts to find the independent “signals” that are being mashed together to create the data as we have them. It does this by finding the most non-gaussian elements, deeming these to be the useful features. This method works well for separating out something like multiple people speaking at once, as each is an independent source. However, the visualization of the first two dimensions does not paint a compelling picture for its efficacy on this dataset with most everything mixed together with no discernible boundaries between classes. The dataset was ultimately reduced to 6 dimensions based on the kurtosis plot above, this number being chosen due to being on par with the other algorithms reducing to about half as well as cutting at about the mean and taking the points above it. The ill performance is not entirely surprising as the dataset is features of wine, so while the features may have been independent measurements they are not necessarily independent features, so breaking them apart may not actually uncover anything.

RP, or RCA for symmetry, operates similarly to PCA, but instead of projecting onto dimensions in order of descending variance, it chooses them at random. Looking at the first 2 dimensions, it is about what one would expect randomly choosing dimensions, the points are fairly mixed without clear class boundaries, but it is possible that they group together in slices of some form when adding in the 3rd or 4th or 6th dimension. As can be seen in the reconstruction error plot below, the error trends downward as the number of components increases (it would be quite something if this were not the case), but does so with no real haste. The variation across 5 runs was only about 5% so the trend is generally consistent indicating that none of the dimensions are really standing out, possibly due to the fact that the difference between the geographical areas is likely to be more of a subtle change in trade offs, rather than an absolute line in the sand after which conditions and results change completely. Because of this, the reconstruction error cutoff was set at around 20%, which kept 10 components, removing only 3. This is not a large reduction, but it is fewer dimensions to have to consider nonetheless and will hopefully help out the later algorithms by giving a slightly more constrained problem space.

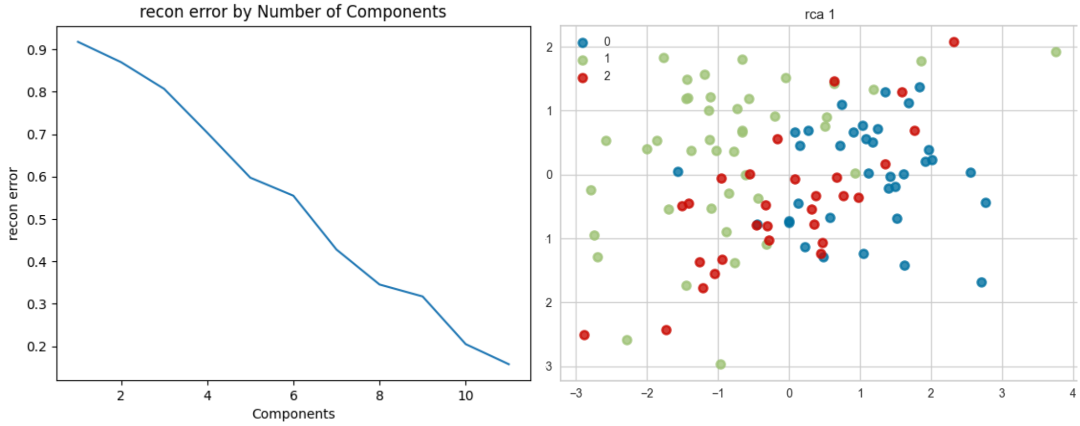

Figure 9, 10—reconstruction error over number of components used (left) and the first 2 dimensions for RCA (right)

LDA did the best in terms of creating isolated spaces of points that a human could look at and see relatively well defined clusters. LDA is a supervised method, so from the perspective of the others, it is cheating a bit. LDA ended up settling on 3 components, aligning precisely with the number of classes, using SVD as its solver. As noted, the first two dimensions can be seen to show a great separation of points which align with the labels excellently. It is worth noting that the number of components that LDA can use is limited by the number of starting features, as would be expected for dimensionality *reduction*, but also by the number of classes, making it more suited for multi-class problems. However, the maximum number of dimensions is not the number of classes, it is actually the minimum of the number of classes **- 1** and the number of features (Linear Discriminant). Thus, the two dimensions plotted below are the entirety of the dimensions left, making it all the better that they paint a good picture of separation in the data.

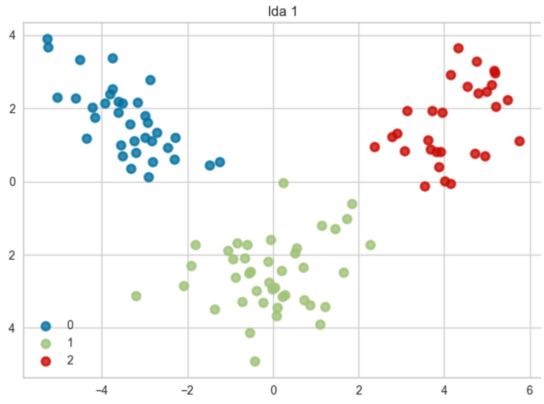

Figure 11—the dimensions for LDA

1.3 Clustering Dimensionality Reduced Data

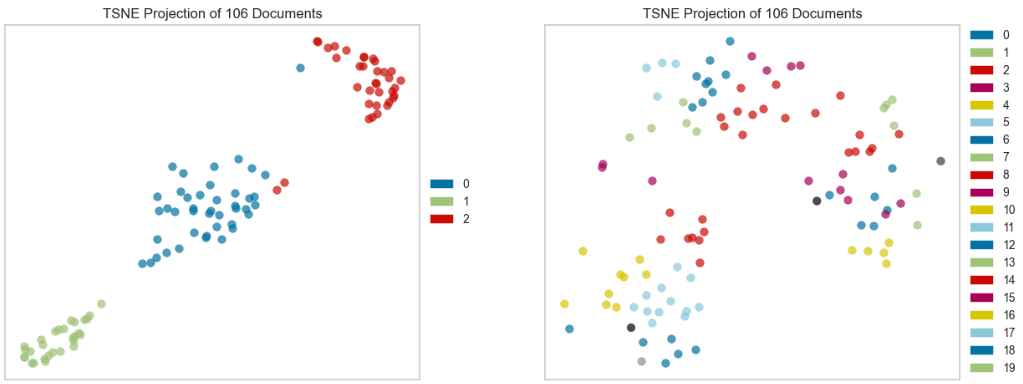

Figure 12, 13—TSNE colored with predicted labels for LDA+kmeans(left) and pca+em(right)

The look of the colored TSNE plots for most of the algorithm combinations is not spectacular. This is predominantly due to the fact that, in some cases, far too many clusters were selected, causing the predicted coloring to more closely resemble abstract art than the true labels. In order of performance of the DR algorithm: LDA + kmeans achieved a perfect coloring with 3 clusters and EM getting a respectable 89% with 6 clusters, PCA with kmeans scored 91% with 3 clusters again and EM scored 73% with 14, RCA performed surprised well given its random nature with kmeans scoring 91% with 3 clusters as well and EM scoring 71% with 20 clusters, and ICA coming up last with kmeans scoring 67% with 2 clusters and EM getting 38% with 5. These results are fairly in line with the dimensionality reduction results, LDA did great as the data were already well separated, PCA had the data fairly well separated, RCA, at least for 2 dimensions, mixed everything relatively evenly across the space, and ICA mostly clumped everything into a mess. Kmeans ended up with the same number of clusters as in its independent run in all cases, except ICA, but with significantly improved scores, again except with ICA. EM ended up with many more clusters than in its solo case, ranging from 5 to 20, as opposed to its independent 3, and actually performed worse in every case, never getting close to its solo score of 95%. This drop could be due to the fact that by removing dimensions, and adding clusters, everything is closer in euclidean space, but also means that all the gaussians are overlapping more causing the soft clustering algorithm to get pulled in many more directions.

2 Dataset 2 - Wine Quality

This dataset contains 11 features describing the quality of wines from the north of Portugal as an integer 0 to 10. The set consists of approximately 6,500 instances that do not cover the classes equally. There are no instances covering the classes 0-2 or 10 and the least populated class, 9, has only 5 instances compared to the most populated, 6, with over 2,800 (Wine quality). This discrepancy leads to a very imbalanced data set where most of the data is contained in 3-4 labels, as it is a fairly narrow version of the bell curve one would expect from a distribution of ratings. The scores were averaged from 3 experts, which helps to explain this large central spike.

2.1 Clustering

The clustering for the second dataset used the same methods as for the first, detailed above. As with the first dataset, the predicted coloring is significantly more visually appealing and makes more sense than the true labels. Looking at the uncolored plot shape, it is hard to determine how many clusters there "should" be but there appear to be 2 or 3 distinct shapes. The tuning for the algorithm ultimately ended up using 5 clusters. The true label coloring seems to indicate a pretty messy space without any good class separation, as there are items scored 5 all over the board. It should be noted that this is not entirely unexpected, these are subjective scores for quality, what is good for one person may be bad for another and there are a lot of ways that something can be "meh." Because of the subjective nature of the scores, it is likely that the space actually looks somewhat like a sea of 5's and 6's with small blips of higher or lower scores as the various elements play more or less nicely with each other. The overall score for this clustering was about 60%, which is higher than expected given visual comparison.

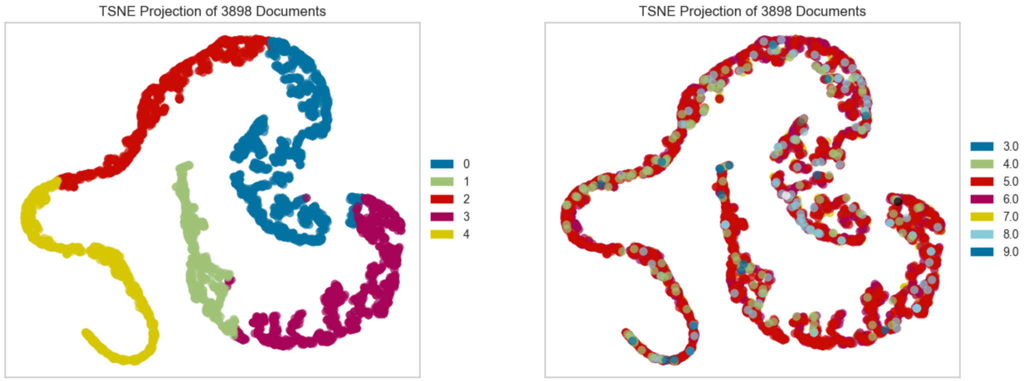

Figure 14, 15—TSNE of the clusters colored with predicted labels (left) and true labels (right)

As with EM for dataset 1, the predicted labeling is a bit messier than for kmeans, but does perform better, getting a score of 63% with 12 clusters. In the same vein as kmeans for this dataset, the shape of the data in the TSNE plot does not present any clear clusters, so it is hard to visually determine how many clusters should be made and where they should be. Similarly, the points rated as 5 in the true labeling cover the entire space with small blobs of the lesser represented classes dispersed around. As noted for kmeans, this is essentially the expected shape of the data in its full space, due to the fact that it is a subjective rating scale. This also means that this seems to be a fairly troublesome problem in terms of clustering, since it is essentially a flat landscape with bumps up or down in random locations, so the classes are not highly cohesive making it difficult to find areas in which a label can actually be determined. It is also worth noting that this is showing that despite the fact that the data labels are distributed along a narrow gaussian, the data themselves are significantly more interwoven.

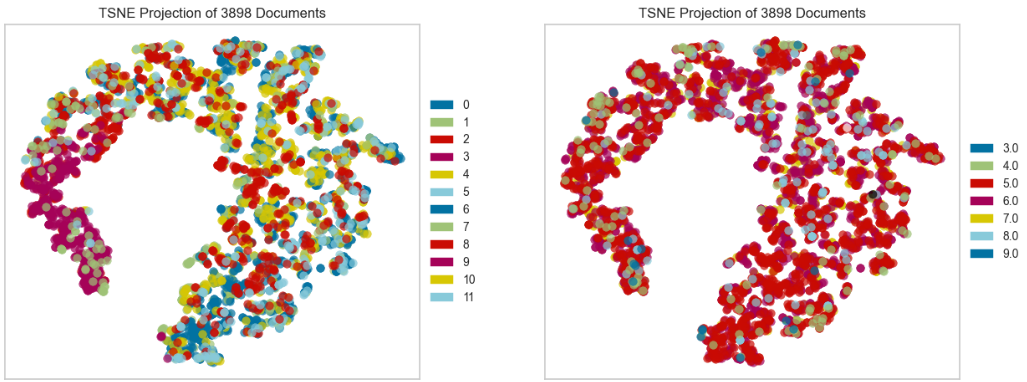

Figure 16, 17—TSNE colored with EM predicted labels (left), and True labels (right)

2.2 Dimensionality Reduction

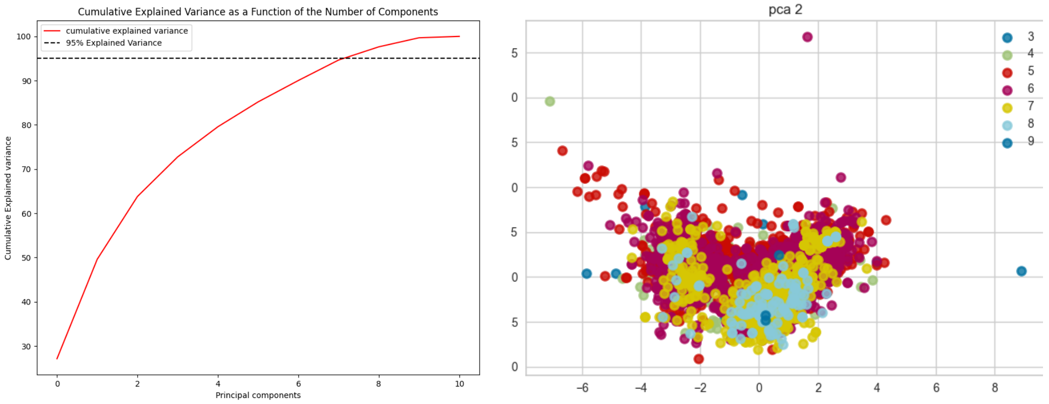

Figure 18, 19—variance over components (left) and the first 2 dimensions for PCA (right)

Using the same variance cutoff method as for dataset 1, 6 components at 90% explained variance were chosen. Looking at the plot of the first two dimensions of the transformed data, it is hard to say that clear areas are differentiated in any meaningful way, especially after the label coloring is applied. As with the few green points in dataset 1, it is possible that these points separate out more cleanly when some higher dimension is applied, leading to more natural areas of grouping, at least to the human eye. One of the likely problems here, as mentioned above, is simply that the scores are subjective, making useful mappings between score and alcohol percentage, for instance, hard to pull out, especially since this may vary across judges, ultimately averaging down to 5 or 6 and complicating any real mapping from features.

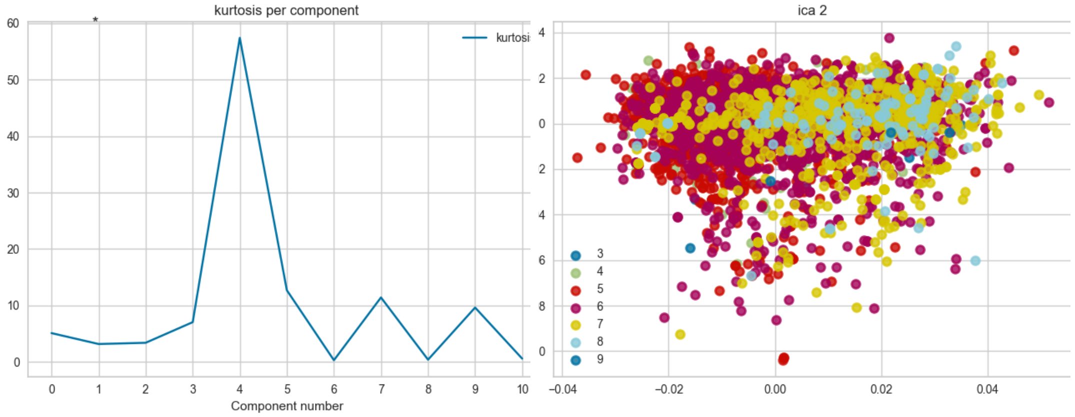

Figure 20, 21—kurtosis per component (left) and the first 2 dimensions for ICA (right)

As with the above algorithms, the same methods were used as were for dataset 1, taking the most kurtotic components from the solution with the highest average kurtosis. As with the first dataset, the first two dimensions of the resulting projection mostly clumps everything together with some outliers, but it is possible that with the addition of the 3rd or 4th dimensions, the points would show something more closely resembling logical groups for the labels. As mentioned above, this is a highly middle heavy dataset, which is likely not helping here, but also the fact that, as with the first dataset, these are not necessarily independent features, which is what ICA would be trying to extract. With these two features, it does not seem surprising that ICA is not performing very well, as this type of data is simply not in its wheelhouse due to the features being entangled and relying on one another.

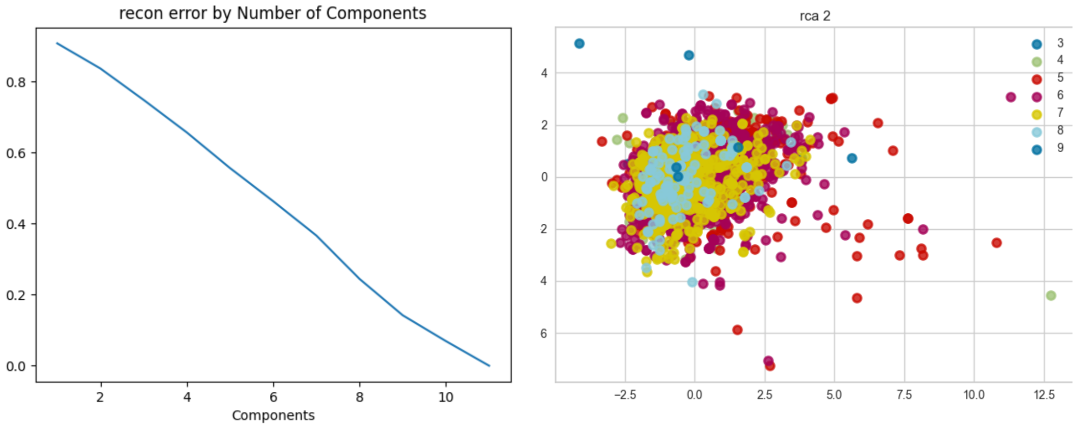

Figure 22, 23—reconstruction error over number of components used (left) and the first 2 dimensions for RCA (right)

Using the same methods as dataset 1, 9 components were selected to get under 20% reconstruction error, with an average standard deviation of 4%. As with the first dataset, there are no discernable clumps of certain classes within the first two dimensions, but this dataset does present everything much more clumped together as opposed to the wider scattering present for dataset 1. As with the other dimensionality reduction algorithms, it is possible that the addition of some of the higher dimensions not easily plotted or visualized would provide better separation of the classes’ data points than is being represented in just the two plotted.

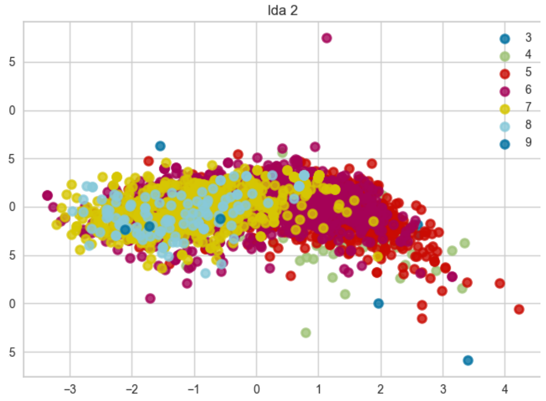

Figure 24—the dimensions for LDA

For dataset 2, in contrast to dataset 1, LDA did not provide amazingly clean class separations. For this dataset it actually reduced the dimensions to 6, rather than 2, keeping the same svd solver, so it is possible that these clear separations would be apparent if 6 dimensional plots were feasible. However, as with all the algorithms run on this dataset, the ultimate plot is pretty messy in that everything is clumped together, further cementing the problem of this dataset, likely all rooted back to the subjective measures and averaging of scores.

2.3 Clustering Dimensionality Reduced Data

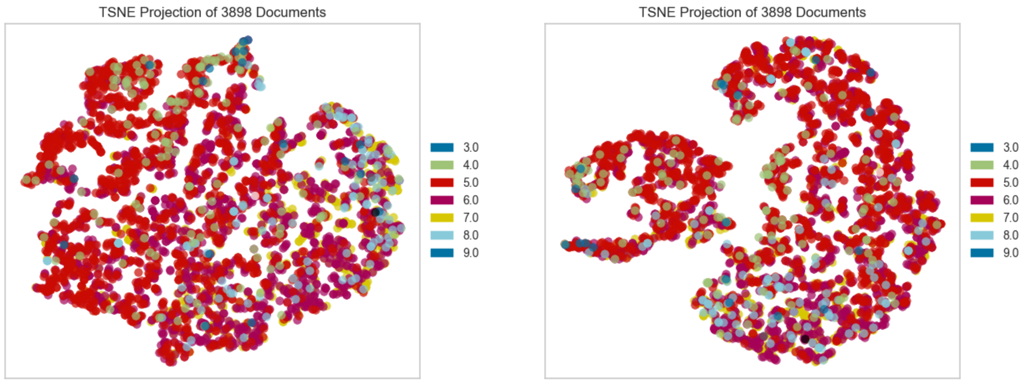

Figure 25,26—TSNE color with true labels for kmeans after lda (left) and pca (right)

As with the first dataset, the colored TSNE plots are pretty much abstract art with the predicted classes all over the plot, with the true labeled ones looking much the same but with a distinct preference for red, as that was always the color of 5 in the plots. PCA and LDA performed the best on average with PCA scoring 61% for kmeans with 5 clusters and 64% with EM with 12 and LDA getting 63% and 62% for kmeans and EM, respectively, both with 7 clusters. ICA scored 61% with kmeans and 65% for EM with 13 and 9 clusters, respectively. RCA mirrored these scores with 58% for kmean and 64% for EM with 4 and 10 clusters, respectively. The number of clusters for kmeans went up on average from the in its solo run while getting slightly better scores, about 2 points higher. EM generally had fewer clusters than the 12 in its solo run and a slightly worse score by about a point. This makes some amount of sense, as noted in dataset 1, where kmeans is aided a bit by having items more closely together, whereas this could actually be messing up the gaussians a little due to having them less separated and, thus, overlapping more.

3 ANN with Reduced Dimensions

The second dataset was chosen for use with the neural net as it was the one that performed worse and had the most to gain from dimensionality reduction. The models were all run to convergence via early stopping and a high max iterations, i.e., one before which all runs stopped. As can be seen from the table of times and scores, no huge improvement in score was achieved, only about 3% better with the best performer, LDA, with some even performing worse. Both of these results make sense, as the reduction of dimensions may be removing some noise, but may also be removing some useful information, or both, resulting in a slightly higher, but nonoptimal, score. However, this is not to say that there was no real benefit to processing the data through a DR algorithm, even the slowest run only took about a third of the time, with the slowest one showing improvement running in 20% of the time and the fastest running in about 13% the time *in addition* to performing 3% better. Domain knowledge about the chemistry of wine could lead to more targeted reduction/better algorithms for preserving these important features, while removing noise.

DR Algorithm	Test Score	Runtime (seconds)
Original	*0.51*	4.44
KMeans	*0.44*	0.79
EM	*0.50*	1.42
PCA	*0.53*	0.82
ICA	*0.53*	1.00
RCA	*0.52*	0.64
LDA	0.54	0.57

Table 1—Dataset 2 test scores and runtimes

4 References

1. Gaussian mixture model selection. (n.d.). Retrieved from https://scikit-learn.org/stable/auto_examples/mixture/plot_gmm_selection.html#sphx-glr-auto-examples-mixture-plot-gmm-selection-py
2. Linear Discriminant Analysis. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html#sklearn.discriminant_analysis.LinearDiscriminantAnalysis
3. UCI machine Learning Repository: Wine data set. (n.d.). Retrieved from http://archive.ics.uci.edu/ml/datasets/Wine
4. UCI machine Learning Repository: Wine quality data set. (n.d.). Retrieved from http://archive.ics.uci.edu/ml/datasets/Wine+Quality