Machine Learning Project 3

Introduction

I re-used my 2 datasets of the Contraceptive Method Choice (CU) (Lim, 1987) and Wine Quality Red and White (WQ) (Cortez, 2009). The Contraceptive data was split into 3 categories None, Short Term, and Long term. In Assignment 1 I analyzed what factors contribute to people's decisions about using contraceptives. I changed it to a binary class, grouped short and long term, and renamed it Contraceptive Used. Wine Quality is rated on a scale of 1-10 but most of the wine only got 5/6. In Assignment 1 I targeted this subset of the data as it was noisy, harder and provided a learning challenge. Contraceptive Used has 9 features and 1473 instances and Wine Quality has 12 features and 4432 instances by combining the red and white data-sets. The data-sets are similar percentage-wise with both Wine Quality and Contraceptive Used around 42.5/57.5 {0, 1} class distribution. This made it interesting to see 2 binary data-sets with similar class distributions but different number of instances. An important difference in the data-sets is that Contraceptive Used has many categorical features with a logical ordering but implicit scaling; whereas Wine Quality is pure numerical data. As I noted in Assignment 1 ”In Contraceptive Used there are many Categorical features. They have a logical ordering but the scale is assumed. For example, Wife's education range 1-4, Husband's education 1-4, Husband's occupation 1-4, Standard-of-living index 1-4. These Attributes have inherent assumptions about the data such as an Undergraduate degree has double the ”distance” than a High School degree and .5 of a Masters. The ordering is understandable but the scale may or may not be true. In contrast, Wine Quality has pure numerical Attributes i.e. 8% alcohol is double 4%.”

The data was first split 80-20 for train-test so that the test set wouldn't leak into the training data then pre-processed with `Quantile_Transformer()`. I used `Quantile_Transformer()` as some features in both data-sets had orders of magnitude in difference. The pre-processing to scale the data is to equate features for K-Means and ANN.

Contraceptive Used: Clustering

For Contraceptive Used I used a variety of methods to determine the number of clusters. I plotted the Elbow Method which suggested a K of 8. However, since there was not such a strong curve to the elbow I went on to plot Silhouette which also suggested 8. When I attempted to plot the data I looked at which columns had the highest Pearson correlation coefficient to the clusters. These turned out to be the columns of Husbands Education and Media Exposure which had correlations of -.44908 and .50101 respectively. The negative sign just means it's an inverse correlation and we can still predict the labels by going in the opposite direction. These 2 attributes add up to 1 (ignoring the negative as we are just concerned about predictive power) and can fully be depended upon to label the data. However, plotting a scatter plot with these labels produced only 8 points – indicating that we had successfully clustered the data but perhaps over-fit to the labels a bit. However, when I explored Expectation Maximization using a BIC score it scored best on 7. This indicated to me that 8 for K-Means isn't such a bad number as K-Means is a subset of EM which

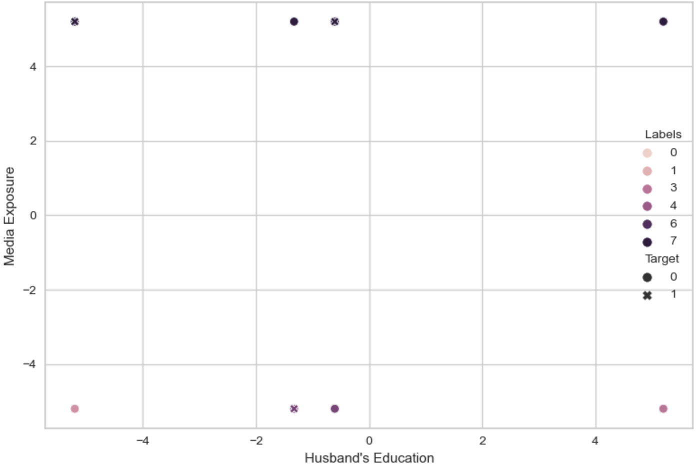

Figure 1: CU K-Means

had a similar number of clusters. Additionally, as Contraceptive Used is a largely categorical data-set it seems that the algorithm is using the full range (1-4) of one feature and splitting the other range (1-4) in half which is indicated by the 8 points/clusters on the scatter-plot. All instances within these 2 features can only have 1 of 8 permutations of values. So, immediately we see problems of a small number of discrete valued attributes in clustering that it is hard to analyze the clusters. This is because they all group together in the number range and can't be differentiated. Since our 2 ranges are both 1, 2, 3, 4 all the data points will be within some combination of those (i.e. 2, 1) and extremely dense with no ability to see the clusters. For EM I used BIC score and cycled through a range of `n_components()` to find the best one. The best score was at 7 and it validated my choice in K-Means of 8 as K-Means is just a more specific version of EM so it made sense they would have a similar number of clusters. We can contrast Figure 1 by plotting the more uncorrelated continuous attributes. Here we can see the data is much more separated as there are more values that the instances can take on. However, the clusters are ill defined as these attributes aren't well correlated with the clusters.

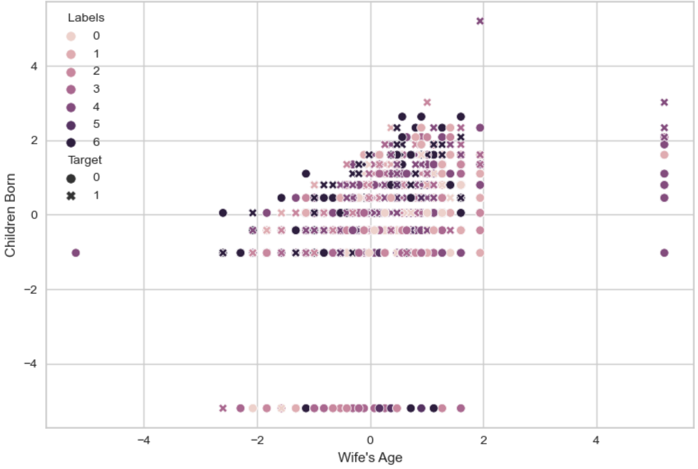

Figure 2: CU GM

Data Reduction

Data Reduction however, helps us tremendously here. It combines the features together to create new features and in this case the new features will be more continuous which helps us analyze them.

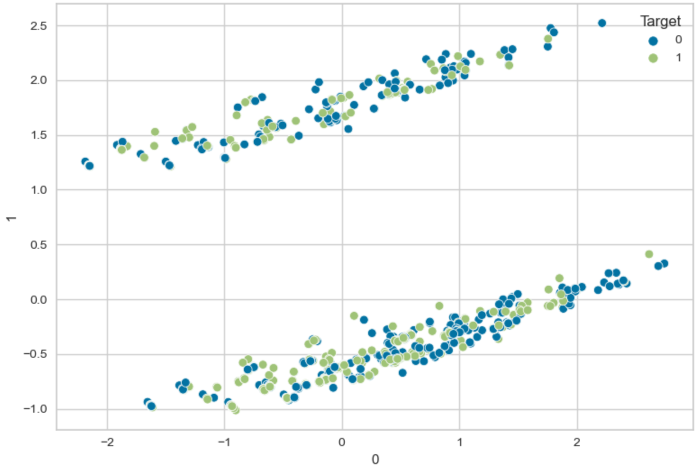

Figure 3: CU PCA

I performed PCA on the data and selected the vectors with the highest Eigenvalues and proceeded to plot them. As we see, by combining features by maximizing the variance between them to create new attributes makes for a much more diverse instance space. As PCA maximizes the variance this creates a more continuous space and spreads out the instances. Choosing the vectors with the lowest variance just produces a mixed up blob of dots. There is still some variance and we see some sections; however, it is not nearly as well defined as the best ones. In PCA we want to maximize our variance while reducing our data and that gap between minimizing data and maximizing variance was largest at 4 components or about .5 of the original data. This still captured .75 of the variance. This is amazing as we can capture a significant amount of the variance by picking the compnents with the highest Eigenvalues in diminishing order until we are satisfied with our result – whether that be space or accuracy.

Similarly, for ICA, choosing the vectors with the highest kurtosis and distance from 0 will produce independent variables. These produce a well separated graph. As our vectors have lower kurtosis the more ill-defined our graphs get. I used the average kurtosis to examine the number of components and noticed something interesting; in general the more components there were the higher the average kurtosis. However, the 2nd to highest number of components had a higher average then the highest. But, when we select the most kurtotic components of the highest number of components we are able to achieve a higher average from the selected components. For ICA too it seems like the best number of components is 4. After the first 4 components we are adding data for a diminishing amount of independence.

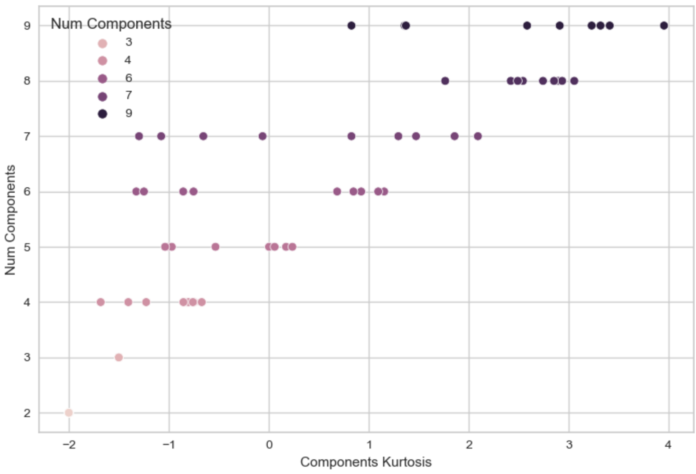

Figure 4: CU ICA Kurtosis

For Random Projection I analyzed the reconstruction error in the data. It seemed to be pretty linear in that as the number of components went up the error went down. This makes sense as the more data we can keep the better we can recreate the data we discarded. As it is linear it is hard to pick a specific spot to pick the number of components. If we need more space saved or speed we can choose a lower number of components and if we need lower error we would go for a higher number of components. In lieu of any other domain specific knowledge I will rely on the domain knowledge from PCA and ICA which both seem to say that 4 components is a good number to reduce to.

I also chose to use LinearDiscriminantAnalysis (LDA) as my 4th dimensionality reduction algorithm. As SKLearn (Pedregosa, 2011) describes LDA that it ”can be used to perform supervised dimensionality reduction, by projecting the input data to a linear subspace consisting of the directions which maximize the separation between classes”. As my dataset is binary data it reduced the data into 2 classes. As my data had a small number of features to start this did pretty well and the misclassifications may be due to noise in the data.

Data Reduction: Clustering

For PCA I plotted the Silhouette score and had over a 20% gain by increasing the clusters to 10 instead of 8 (8 was better too but the score was still going up). After 10 clusters we had diminishing returns on the score. Gaussian Mixture also has 10 for its components which reinforces that K-Means is a subset of Expectation Maximization. This amount of clusters are close to the previous amount of clusters and make sense. Previously we had 8 clusters but the data overlapped and was to dense to make more clusters (see Figure 1). PCA was able to recombine the data and increase variance creating a richer instance space and more possibilities for recombination for the clustering algorithms to choose from. It still has that upper and lower bound as seen in Figure 1 and 3. within these It could be that the large amount of features that only have a range of 1-4 present a hard problem for

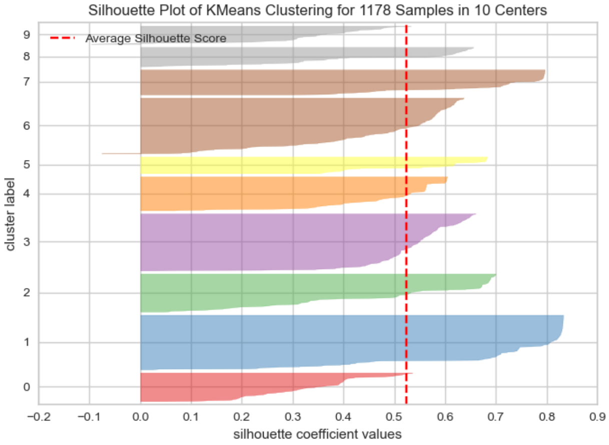

Figure 5: CU PCA SIL

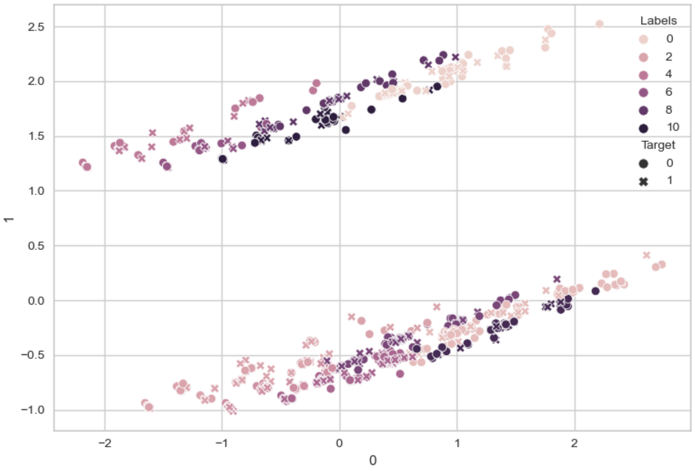

Figure 6: CU PCA K-Means

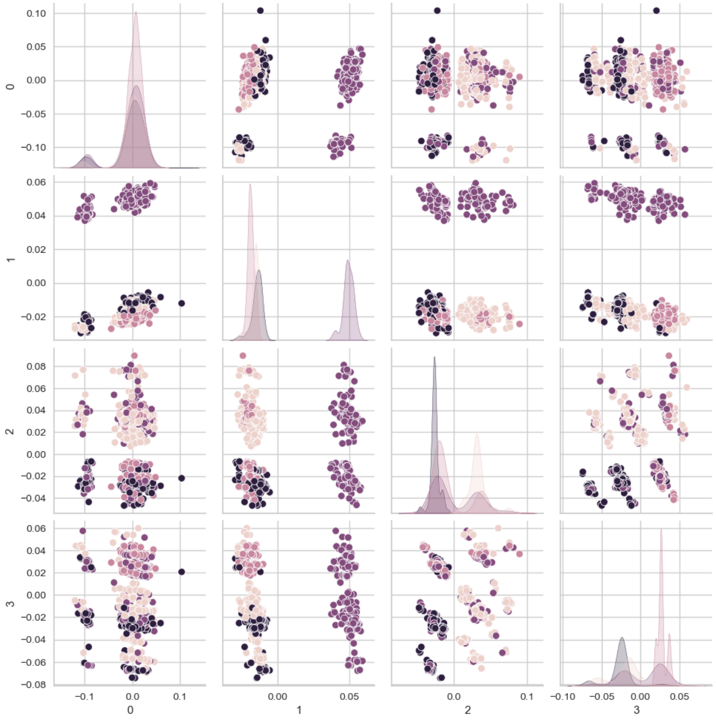

Figure 7: CU ICA GM PP

PCA. Since PCA attempts to maximize the inter-feature variance if the actual features have low variance this can be hard to maximize. Additionally, these may be the orthogonal lines that PCA projects. It is interesting to see that the clusters seem to be on a line which also validates that PCA projects the data on an orthogonal line and the clusters adhere to that projection. As we continue to plot components with lower variance the data starts being ill defined and the labels lose cohesiveness.

ICA however, does a better job of showing us these clusters. K-Means produced 8 clusters while Gaussian Mixture produced 4. Perhaps Gaussian Mixture soft-clustering produces a smaller number of clusters due to its lack of ”absolute value” method of clustering which may allow it to reuse old clusters to explain the data. Soft-clustering may enable it to explain the data with a conglomeration of labels thus voiding the creation of new ones. This ICA pair-plot has 4 labels from Gaussian Mixture. As we see the purple cluster is the best defined across the kurtotic components. However, as we progress along the dimensions other clusters begin to appear. 1st Beige in the second row/col, then black starts in the 3ird. This graph allows us to peer through the dimensions and see the multi-dimensional clustering that these algorithms are picking up on. ICA is similar to PCA in that less kurtosis leads to less defined clusters as we see in the bottom right of the graph. The more kurtotic the components are the better defined the clusters. We see similar patterns here to the unreduced data. Just like the original data had very distinct clusters with one line on top and another on bottom we see similar patterns here. It may not be 2 horizontal lines but there are 2 halves of the data. This gets less clear as we add components and the data becomes homogeneous.

In Random Projection the K-Means Silhouette suggested 4 yet for Gaussian Mixture the BIC score suggested was 8. This was the first time that K-Means was less than Gaussian Mixture. Additionally, the clusters seem to be hovering around numbers like 4 and 8 (Gaussian Mixture pre-reduction was 7). This can be for a number of reasons. 1) The underlying data is categorical and the categories come in either 2s or 4s and the clustering algorithms are seeing this even after the data reduction. 2) The underlying data is a binary classification data-set which seems to be divided into sub-clusters which explains why the numbers are 4 and 8. Here we see some of the orthogonal lines similar to PCA but not as well defined. That is because Random Projection is ran-

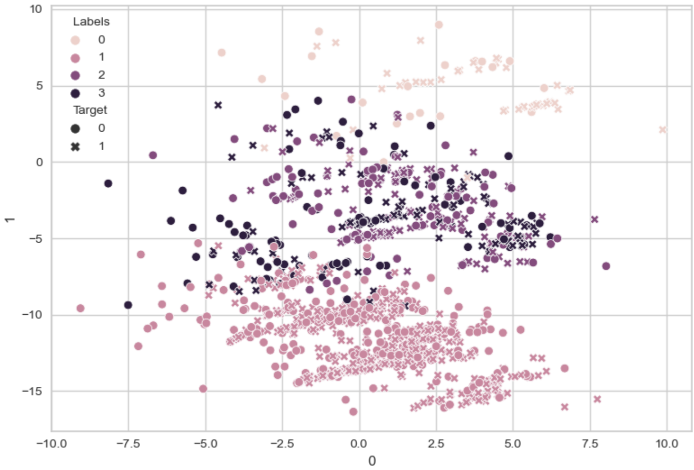

Figure 8: CU RP K-Means

dom and doesn't choose the best components so it will be less orthogonal. Yet, it still does pretty well. The clusters are well defined but don't seem to have any correlation with the Target values. This goes against our earlier hypothesis that the sub-clusters are multiples of 2 due to the binary Target. However, perhaps in unseen higher dimensions it would be clustered differently so the disproof is not absolute.

LDA clusters the data and reduces the dimensions based on the Target. It does pretty well seeing how it ends up in only 2 dimensions. It also gives us a confidence score which we can use to show how LDA rates each instance of the data. As expected, K-Means and Gaussian Mixture also give us 2 clusters based on the reduced data.

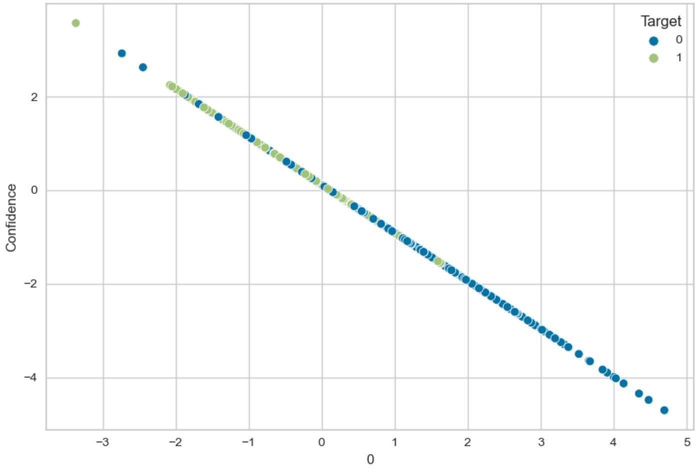

Figure 9: CU LDA

Wine Quality: Clustering

In contrast to Contraceptive Used the Wine Quality data-set is less discrete in its attribute values. This becomes immediately apparent when plotting the clustered data using Gaussian Mixture. Besides being less discrete the data-set also differs in a number of other ways. 1) We can see that the Target values do get semi-separated along the Alcohol axis. 2) The near continuous nature of the data allows it to be more Gaussian. However, this is also a binary data-set and both Gaussian Mixture and K-Means are multiples of 2 (4 and 8 respectively) which is evidence that binary data-sets can be sub-clustered. I used BIC score for Gaussian Mixture and a combination of Elbow and Silhouette scores to inform my choice of clusters. Note the Gaussian nature of the data. This is important because as we progress through the data reduction techniques we will see how they affect the data.

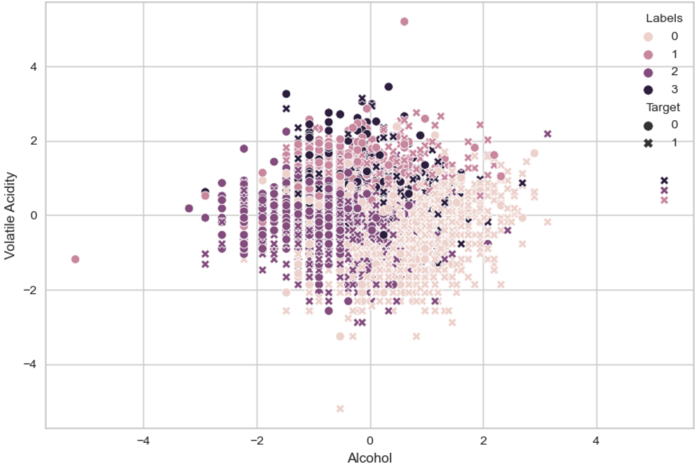

Figure 10: WQ EM

Data Reduction

I believe a most interesting graph is for PCAs 2 components with the highest Eigen Values which produced this graph. This contrasts beautifully with the Contraceptive Used data in which the data points seemed sparse and the components didn't seem to correlate to the Target values. Contrasting to the Wine Quality data allows us to see how big of a role the underlying data plays. Here the underlying data was densely clus-

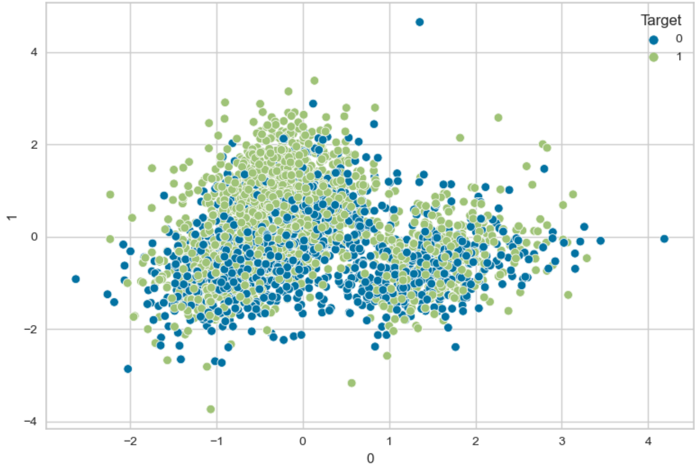

Figure 11: WQ PCA

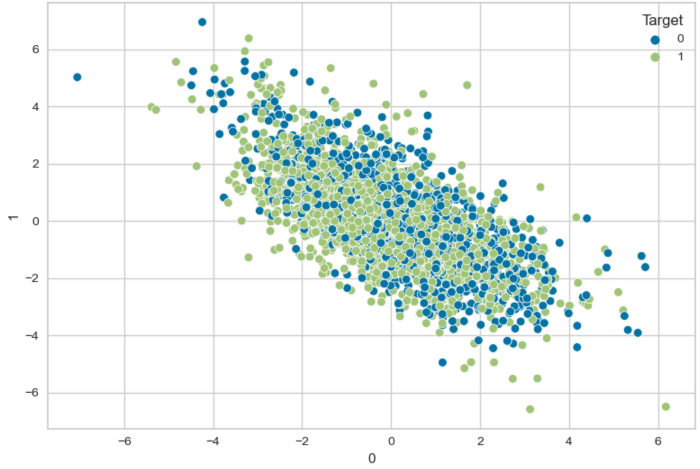

Figure 12: WQ RP

tered, Gaussian, and better correlated to the Target values. PCA can only work with what is there and so the resulting graph is also densely clustered, Gaussian, and better correlated to the Target values. Its shape recalls the original data but PCA introduces those orthogonal lines again. I chose to maximize the difference between space reduction and variance which gave me 5 components – down from 12. For ICA the highest number of components gave me the best average (unlike Contraceptive Used) and I reduced it to a subset based on diminished returns of kurtosis/space. This reduced it to 4 components. If we order the components by their kurtosis and plot a pair-plot the Target will be best defined by the best components and decrease as kurtosis and independence decrease. For my Reconstruction Error as it was just a linear increase the more components I removed I relied on domain knowledge of the PCA and ICA components. Plotting the Random Projection gave this graph. As we see it separates the Target decently but what surprised me is that it is more orthogonal than PCA. Perhaps the difference is because PCA chooses its components with maximum variance and that led it to Figure 11 despite it not being so orthogonal. RP randomly chooses its components and then maximizes variance along the orthogonal plane. LDA performed amazingly well and since GM and K-Means both produced 2 labels affirming the Target we will talk about it in the next section.

Data Reduction: Clustering
LDA has a huge advantage in data reduction by seeing the Target labels. This is also a disadvantage if we don't have those labels it can't be used. For this circumstance in which we can use the labels it is a huge advantage. Both Gaussian Mixture and K-Means produced 2 clusters. As seen

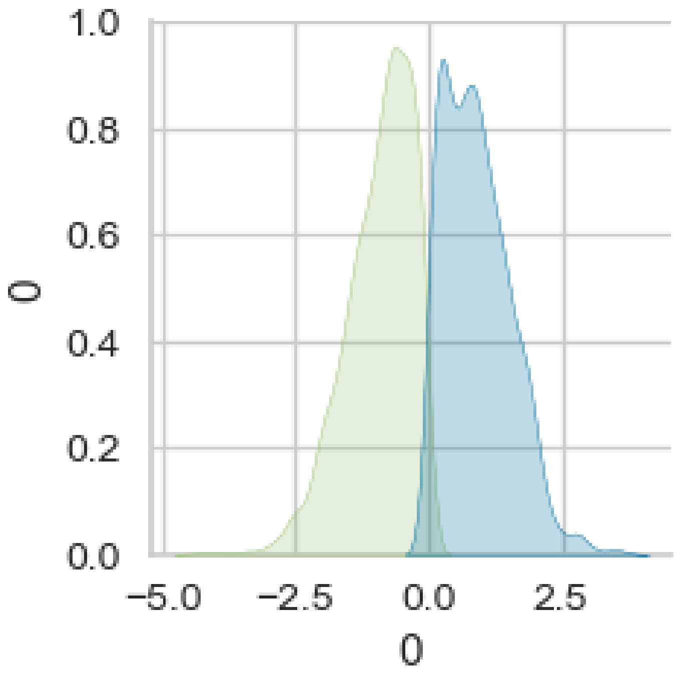

Figure 13: WQ LDA K-Means

LDA doesn't care about being Gaussian, all it cares about is splitting the instance space into the Target sub-classes in a lower dimensional linear space. (Perhaps the more flexible Quadratic Discriminant Analysis would be able to further separate the data.) I believe the overlap in classes is unavoidable due to the nature of the underlying data. As we are talking about ratings (Wine Quality) there will inevitably be some overlap – especially since we took the middle and hardest ratings (5/6) to separate.

PCA with Gaussian Mixture produced 5 clusters using BIC score. As the data was reduced and seemed to produce 2 clusters I would expect those clusters to be subdivided into 4. However, it produced an extra cluster in the center of all the other clusters. Perhaps this is due to soft clustering. If there is a big enough group of data that can belong to many clusters it will create an ”uncertainty cluster”. This cluster would comprise of data that has a high probability of belonging to any cluster. In 13 we saw some overlapping data that couldn’t be separated and maybe Gaussian Mixture is picking this up and making it into a new cluster. It is interesting to see the soft-clustering here. In general the 2 sub-clusters (light/dark) seem only to soft-cluster with each other – light with light etc. However, the middle cluster overlaps with all of them. Is this a condition of its local or can it be that the middle cluster is a cluster of clusters. It extends well into light and dark territory. However, this can also be due to higher dimensions.

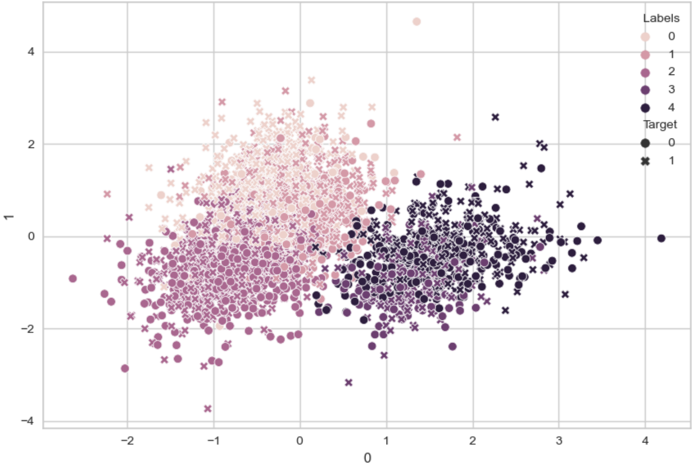

Figure 14: WQ PCA GM

In our Random Projection K-Means we also get interesting results. When I plotted this for Gaussian Mixture the clusters were very intertwined even though the BIC score was for only 4 clusters. Here with 8 clusters via Elbow and Silhouette it is much more distinct. This again highlights that Gaussian mixture and soft-clustering leads to more intertwined clusters despite having a lower number of clusters.

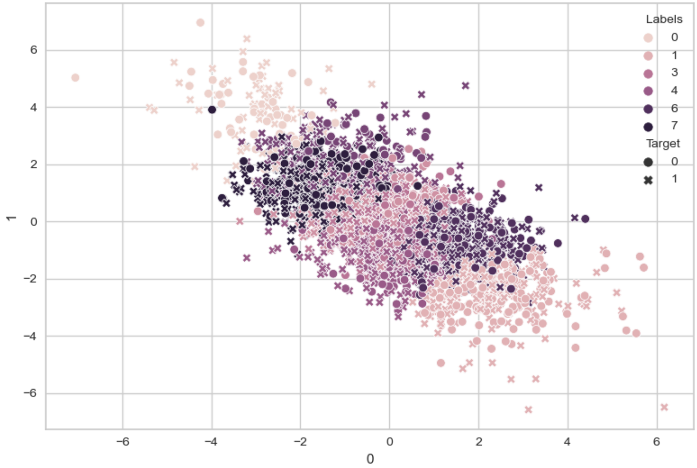

Figure 15: WQ RP K-Clusters

In ICA our K-Elbow score was for 8 also. While it’s not the greatest elbow it is also backed by a Silhouette score validating it. Additionally, we have domain knowledge that indicates our data is subdivided into 4s or 8s due to the binary Target labels. It is fascinating that the clustering algorithms

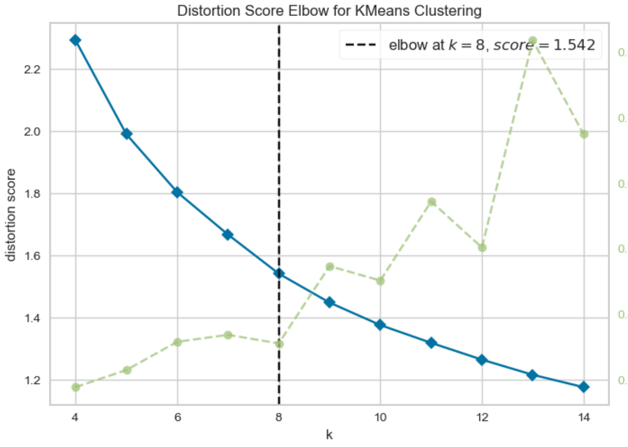

Figure 16: WQ ICA K-Elbow

consistently pick up on this despite not having access to the Target labels. We see the same pattern repeated again in ICA as Gaussian Mixture produces 5 clusters (near 4). This seems to play into the ”extra” soft-cluster again in the plot that Gaussian Mixture seems to produce. We can see 4 distinct clusters (we had 4 originally too) and an extra one hanging out at the edges. The data definitely looks less Gaussian than our original data in Figure 10 which is expected as we maximize kurtosis in ICA.

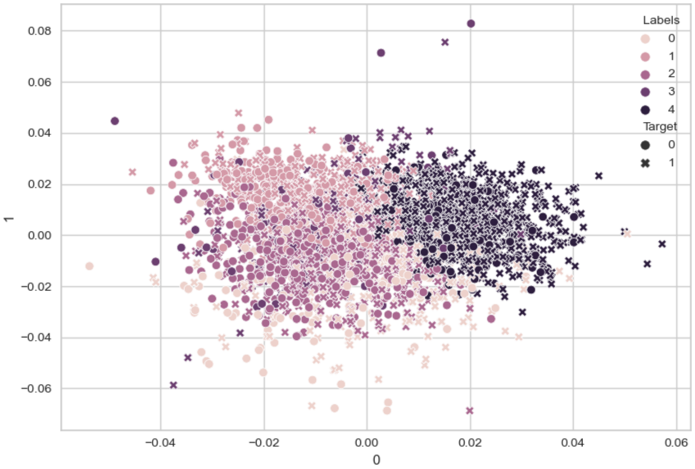

Figure 17: WQ ICA GM

Data Reduction: ANN

I passed in all the transformed data per reduction to ANN instead of the original training instances. I then fit the data and plotted the loss/iterations. Surprisingly, ICA did extremely poorly. In my graphs (not shown) it seemed to separate the data well but wasn’t really interesting. Perhaps the nature of the underlying data and as ICA moves away from that the Target values become harder to predict. LDA in my opinion is the underrated factor here. It is extremly fast to converge and produces surprisingly good results despite it only having 2 components. This can be attributed to LDA being able to see the Target data which gives it an unfair advantage over the other algorithms. PCA and RP perform similarly which is expected. PCA performs slightly better which is surprising as it gets to pick the best components whereas Random Projection doesn’t. If one views accuracy as more important go with PCA; if speed is more important Random Projection is better. The interesting thing here is that PCA performs worse in the beginning. As the Neural Net re-weights the components PCA does better. This is likely due to PCAs higher variance which enables the Neural Net to better separate the data. Both PCA and Random Projection ultimately perform better than LDA as they haven’t reduced their data to the same extant and have more attributes. This allows for a richer expression of the instance space which can be translated into a better score by the Neural Net.

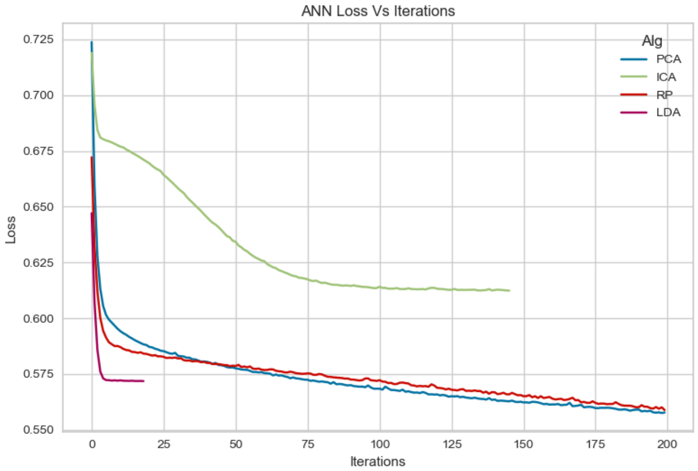

Figure 18: ANN P4

Scoring the transformed test data which I separated before the data reduction algorithms also produces interesting results. The final results with Sklearn ANNs mean accuracy score were interesting. From greatest to least, LDA: 70.91, RP: 67.86, PCA: 67.74, ICA: 55.35. So despite not fitting well to the classes LDA performed best of all. I attribute this to the fact that it can see the Target labels and tries to maximize the variance between the classes. ICA did worst, the only surprising thing is that it did much worse. Again, perhaps the underlying data is Gaussian in nature and finding independent components is hard if they don’t exist. The other interesting thing is that Random Projection did *better* than PCA. An explanation may be that PCA maximizes the variance to the seen training data and that variance doesn't translate to the unseen data. This makes it difficult for the Neural Net to achieve a good loss with a lower than expected variance. Random Projection by nature won't overfit the variance to the seen data and this ends up doing well on unseen data since it will also have a high variance.

Clustering: ANN

I generated the cluster labels for the Wine Quality data, OneHotEncoded them, and ran ANN on the Encoded clusters. The clusters for K-Means and Gaussian Mixture were 8 and 4 respectively. Doing so produced this graph. It performed surprisingly well. In fact, it has similar results to ICA here but without the data complexity. It also is extremely fast iteration-wise due to the simplicity of the data. It works because the clusters are a form of encoding the data in a different form and assigns a label to it based on how it is grouped in the higher dimensional space. My final scores with Sklearn ANNs mean accuracy on the test data were K-Means: 60.87, and Gaussian Mixture: 58.96. It is amazing how they performed so well by dividing the data into clusters and just using the clusters instead of the data. In fact they performed better than ICA which implies that sometimes clustering can be used as a data reduction technique in and of itself. Also, due to the simplicity of the data it takes very few iterations to achieve convergence. This can be essential when speed is necessary.

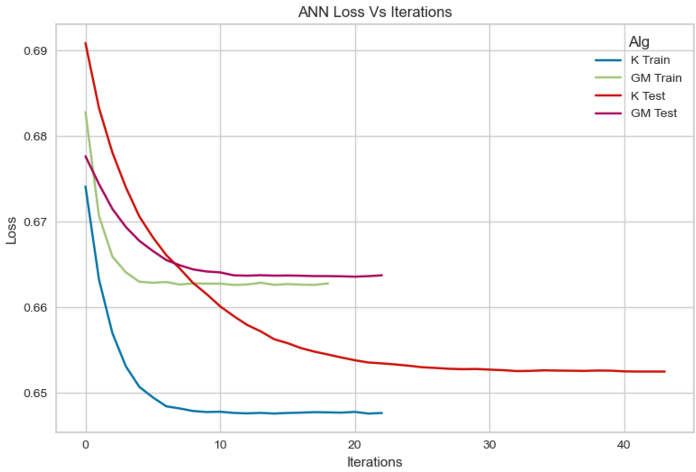

Figure 19: ANN Train/Test Labels

Summary

We saw how clustering can pick up on underlying patterns in the data as it repeats it's clusters of being close to 4 and 8. I believe this is correlated to the binary Target labels of the data and the clustering algorithms are picking up on subdivisions within the data. We also saw how data reduction techniques transform the data. Each algorithm has its own pattern but they are also bounded by the underlying data. For example, in PCA we saw how it reduces its data onto orthogonal lines. In Contraceptive Used however these lines were much clearer vs Wine Quality the lines were more blurred. This is due to the underlying data. In Contraceptive Used the data was extremely discrete but Wine Quality's data was extremely continuous and Gaussian. Therefore, in Contraceptive Used PCA created sparse lines on which we were able to see how the clusters actually followed the orthogonal line. In contrast, Wine Quality's PCA the lines were very blurred. We noted how average kurtosis can be used to hone in on how many components to use for ICA and how we can further refine the reduction by using the most kurtotic of those components. In Random Projection we used reconstruction error to pick our number of components. As very often this error is linear to the number of components some other metric or domain knowledge can be used to finalize the number of components. We used K-Elbow, Silhouette and BIC score for our clustering.

Another interesting thing noticed was how Gaussian Mixture often seemed to produce an extra label and in general also produced fewer labels than K-Means. We hypothesized that the fewer labels were due to Gaussian Mixtures ability to combine the old labels to continue to explain the data as opposed to having to come up with new labels. However, it was not immune to being forced to coming up with new labels. It seemed that if there was a significant amount of data

Gaussian Mixture would come up with a label as an ”outlier” label. We also saw how well data reduction performs and how Random Projection can outperform PCA as perhaps it doesn’t overfit the variance to the training data. Lastly, we saw how we can reduce the data to the clusters and just use those for ANN.

References

Cortez, P. (2009). Wine Quality. Retrieved from UCI ML Repository: https://archive.ics.uci.edu/ml/datasets/Wine+Quality

Pedregosa, F. a. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 2825–2830.

Tjen-Sien, L. (1987). Contraceptive Method Choice Data Set. Retrieved from UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice