CS7641 Assignment 3: Unsupervised Learning and Dimensionality Reduction

Steel Ferguson
Sferguson42@gatech.edu

Introduction

In this assignment I use the data sets from assignment 1 (summarized below) to experiment with clustering and dimensionality reduction. I use KMeans (KM) and Expectation Maximization (EM) to cluster. I use four dimensionality reduction (DR) techniques: Principal Component Analysis (PCA), Independent Component Analysis (ICA), and Random Projection (RP), and Recursive Feature Elimination (RFE). I combine dimensionality reduction methods with clustering for both data sets. I also incorporate dimensionality reduction and clustering into the Neural Network classification model from assignment one. Following is my analysis and findings.

Datasets and Process

To select the two data sets in Assignment 1, I sought data sets that would be sufficiently different in three ways: 1) different relative strength on five supervised models, 2) one balanced (roughly equal rows in each category) and one unbalanced, and 3) and one set with many more rows than the other.

The Pima Diabetes data set has medical information as features with an indicator of diabetes as the label. It is roughly balanced and is relatively small (500 positives for diabetes out of 768).

The Amazon Reviews data set contains data about the review and title texts with an indicator of a low rating as the target (using 1-2 as low). It is unbalanced and large (4k low out of 77k).

The data sets showed different relative strengths on supervised learning models.

I first applied KM and EM clustering to the two data sets reviewing metrics (discussed below) to determine an appropriate number of clusters (k) for each model. I used Euclidian distance as there was not strong indication of a different similarity metric being appropriate based on domain knowledge. After selecting k, I analyze the clusters without the labels and then see how the clusters compare to the labels.

I select the number of components for each type of dimensionality reduction (discussed below) for each dataset using different methods depending on the reduction methodology. I then use the reduced dimensions to create new clusters.

Finally, I used the Pima dataset to measure Neural Network performance using the dimensionality reduction and clustering.

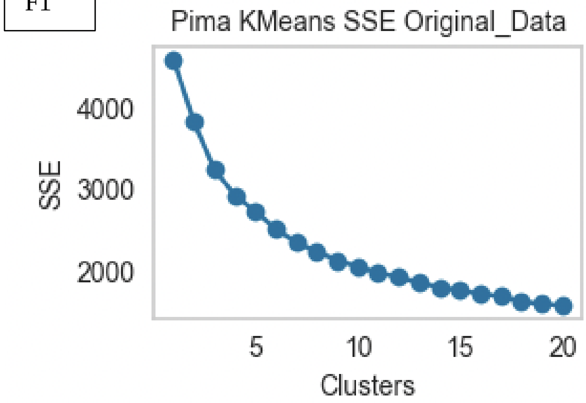
In order to have similar result to compare, I tune each component for the specific permutation before producing artifacts to analyze.

Clustering: KMeans

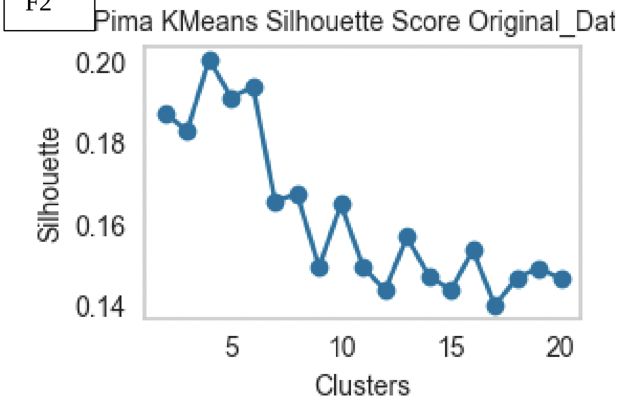
Selection: I attempted finding the “elbow” when charting the Sum of Squared Errors (SSE) by the number of clusters, but for both data sets there was not an easily detectible change in the slope of the SSE line, even with tuning. This can be evidence of the lack of a good natural segmentation (e.g. this data is not from different species of irises that can clearly be clustered “correctly”).

For both data sets I instead used the Silhouette score which considers both intra- and inter-cluster distances and give a quantity for how much the points are closer to their own cluster than how much they are closer to points in other clusters (with low being good). I also used the average of a number of iterations as randomly the clustering methods can get stuck in local optima. I also charted time (not shown) and found that more clusters took more time somewhat linearly.

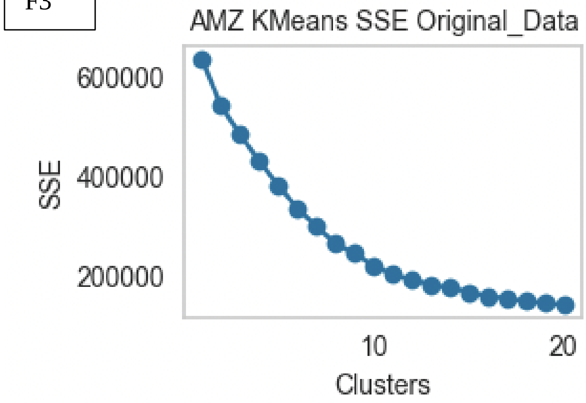
F1



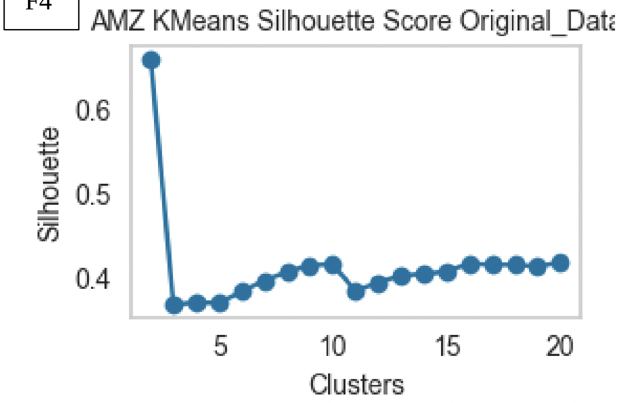
F2



F3



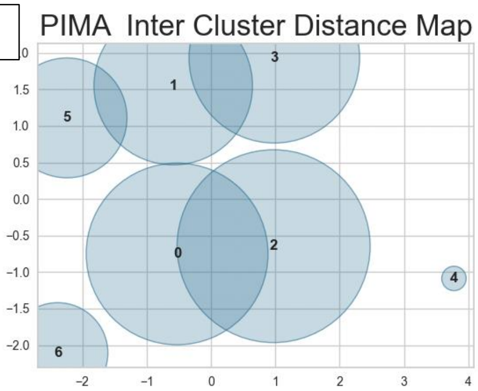
F4



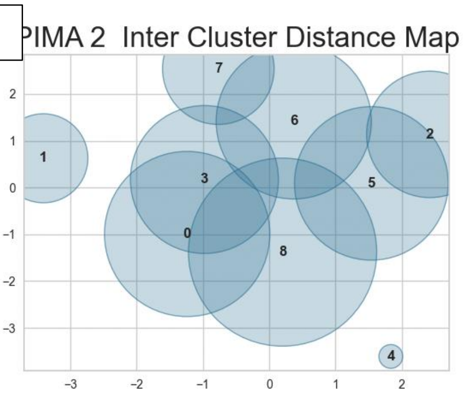
Analyzing: Based on the silhouette charts, I choose k=3 for AMZ and k=7 for PIMA (as it looks like it is the beginning of the "elbow" shape), but I also looked at result when adding two additional clusters to understand what is happening with the cluster formations. I found Yellow Brick's inter-cluster visualization helpful (F5-F8). It preserves the distance between clusters and represents that in 2D. I found F7 and F8 interesting, especially considering the flatness in =2-k=4 in F4. F7 shows a meaningful distinction between clusters (e.g. really long reviews with a high number of negative words vs the typical reviews). When looking at F7 and F8 it looks like F8 creates "unnecessary" clusters as the clusters seem to be right up on each other and that makes sense as we see no large improvement to silhouette score (which considers cluster distances) in k=2-k=4.

We see something similar in F5 to F6, but F2 actually shows some improvement, so we would expect the additional clusters to be more meaningful. We do observe that the clusters in F6 are tighter but not right next to each other as in F8.

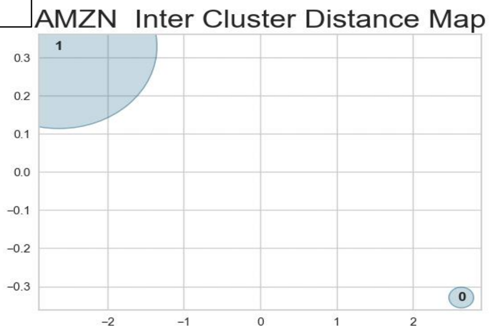
F5



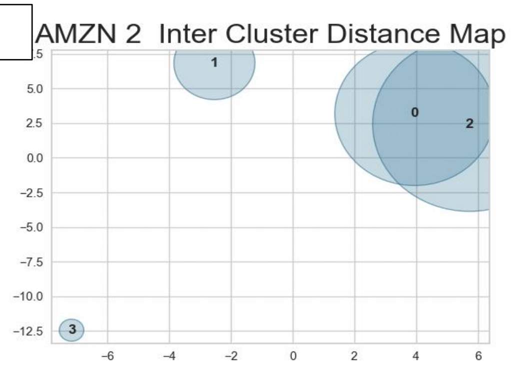
F6



F7

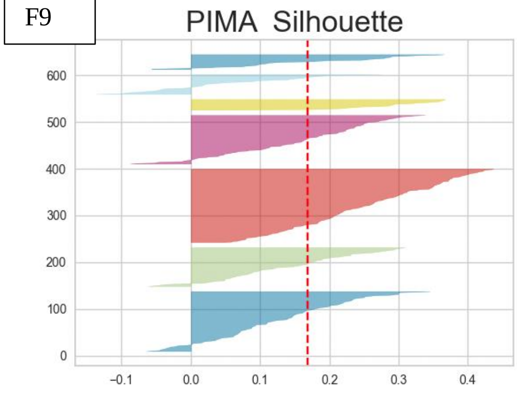


F8

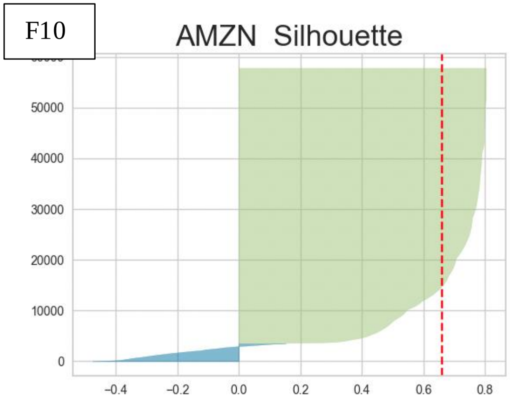


In visualizing the silhouette score at each cluster, we can see that the Pima clusters still have a good deal of cross over. There are data points that are closer to point in other cluster than in their own clusters (points jutting out to the left) for each cluster except for the largest (red) one. The Amazon chart if very interesting as it shows a very large score for the large cluster and a negative score for most points for the smaller cluster. Part of the large positive score should be contributed the fact that it has so many points in that first cluster. There seems to be a natural division in the data that basically cuts out some very dissimilar points (show in F7).

F9



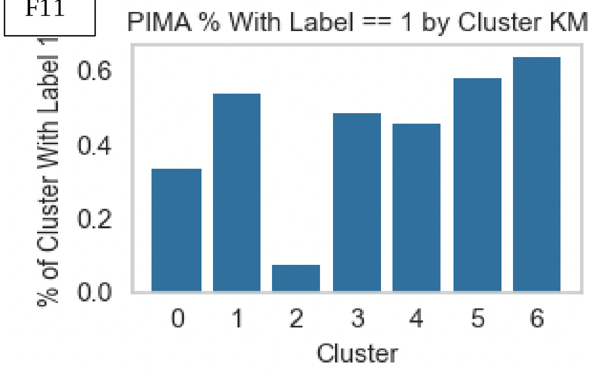
F10



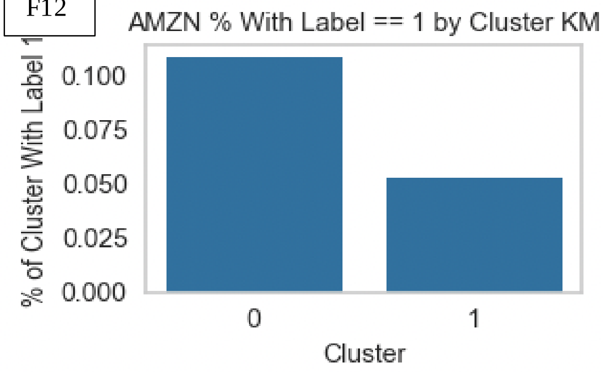
The Amazon cluster does find a meaningful split, and we also can see that the smaller cluster seems to be a subset of the negative reviews as F12 shows 100% in the “negative rating” label. I would interpret this as 1) most reviews are similar (e.g. not containing many negative words) 2) a few reviews are dissimilar (e.g. very long with many negative words) and 3) all of those are actually low rating reviews.

The Pima data set does not have as clear a division, although cluster 2 seems to be mostly of one label. Other clusters group by things that do not line up well with the labels.

F11



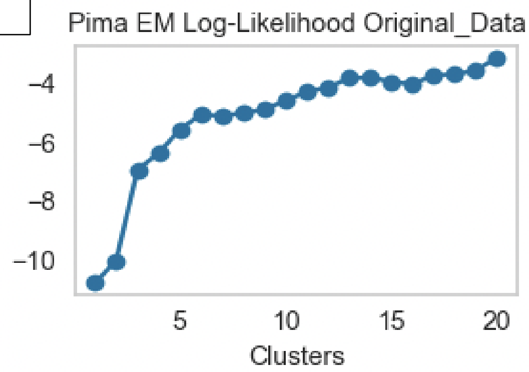
F12



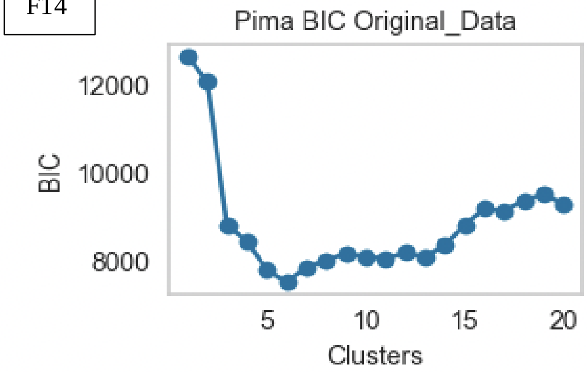
Clustering: EM

Selecting: In selecting the number of components for the EM, I use BIC as it looks to maximize Log Likelihood (LL) (F13, F15) while penalizing for more complexity (higher k). I used BIC instead of AIC as BIC penalizes more and I would like to use the results as a dimensionality reduction later. Pima showed the lowest BIC at k=6, and I interpreted k=4 as the “elbow” for Amazon. Again, time sloped positively with the number of clusters (not shown) as more cluster means more possibilities to explore for the algorithm.

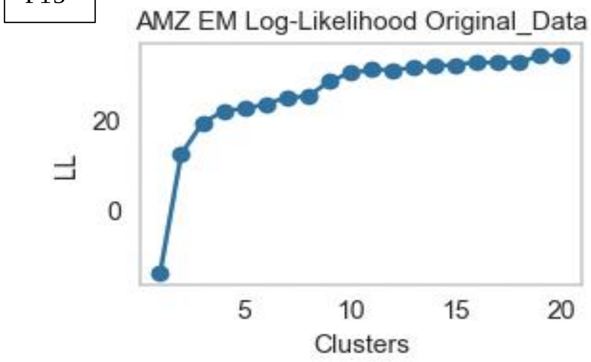
F13



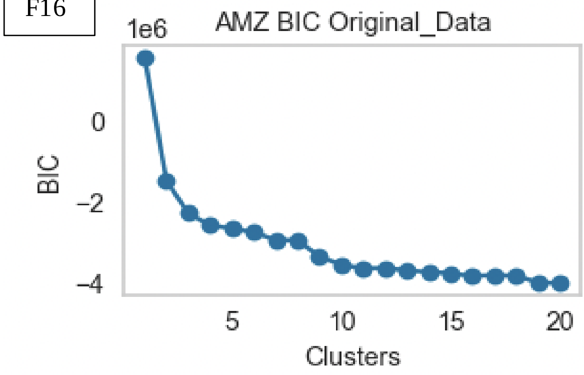
F14



F15

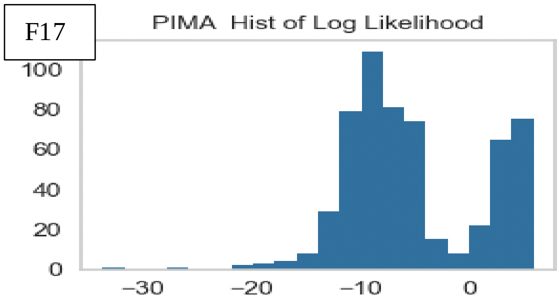


F16

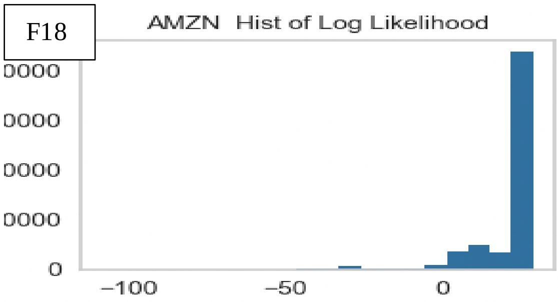


Analyzing: I looked at the histogram of LL to see if I could understand where the points were in relation to the component means. Amazon shows the majority very likely (very close to the means), similar to what we observed in the KMeans. For Pima there was a group with positive likelihood but then another large group with low likelihood (so the log is negative). Based on that is seems that Pima is not easily split into clean clusters, even with some tuning.
For Pima, however, the EM did show a clear "best" answer using the BIC, so I would expect some meaning in having the 6 selected clusters using EM.

F17

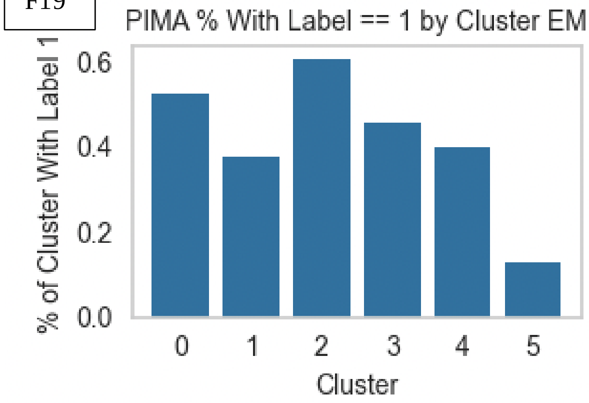


F18

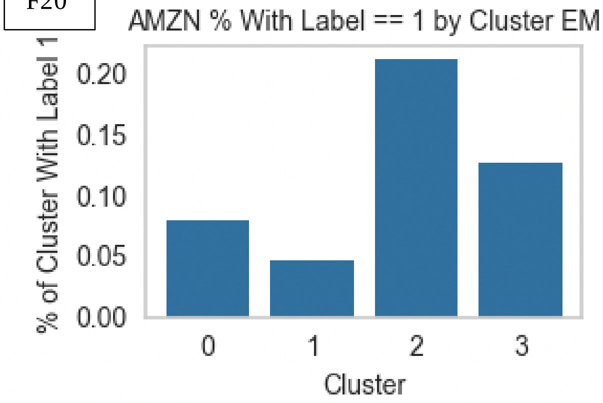


Pima again has one cluster (5) that has lower percentage of the label with all other cluster capturing things that may not line up as well. Amazon does not have the label alignment with these cluster as each has a small portion of the negative reviews.

F19



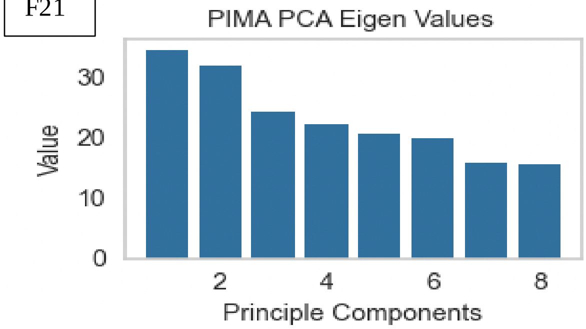
F20



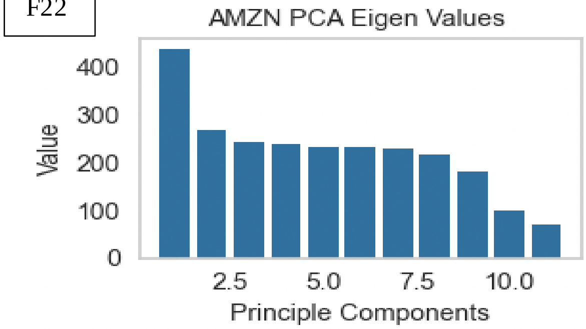
Dimensionality Reduction: PCA

Selecting: PCA creates components that explain variance in descending order so we would expect the Eigen values (and the ratio of variance explained (not shown below)) to show a negative slope by the number of components. We observe that for both data sets (F21, F22). The first two component for Pima and the first component for Amazon have much for explaining power than the rest. The values slope down and then drop off in the last couple. I chose to take all components up to these drop offs (PC=6 and PC=9) as there still seemed to be a lot of explaining power the model would miss out on without that middle section (between the high and low). The PCA will then project the same data point (or close to it) into a new space.

F21

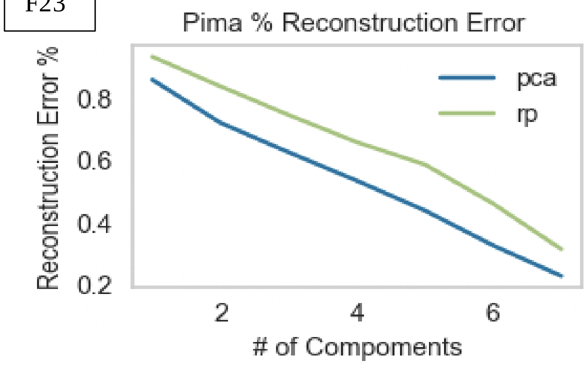


F22

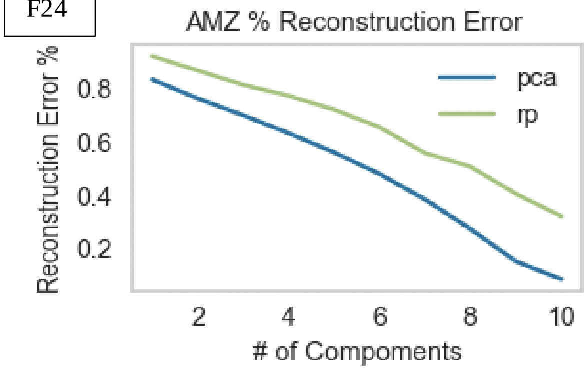


Analysis: The PCA is able to capture more variance and create less error when reconstructing than if the projections were random as it selects the most important (explaining the most variance) first and then takes orthogonal dimension after that. We see in F23 and F24 that PCA has lower reconstruction error (distance between points when projected and then projected back) than its random counterpart. Not shown in the chart, however, is the increase in time. For Amazon, it was 8-10 times longer with PCA (0.19 seconds vs 0.002). Neither data set has a huge number of columns, but we expect that time component to be more important for large numbers of features.

F23



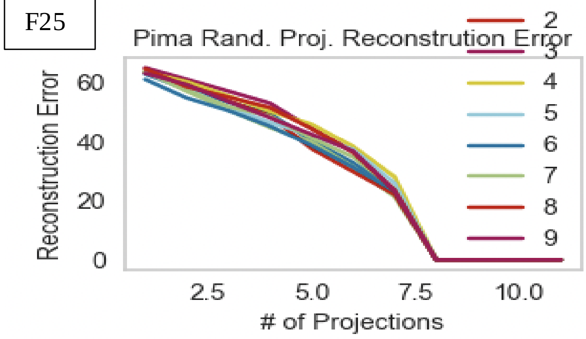
F24



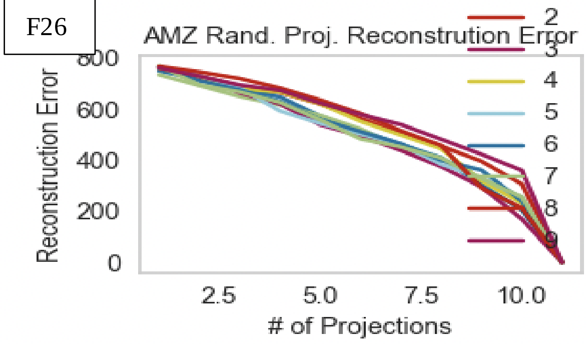
Dimensionality Reduction: RP

While PCA systematically picks the "best" components first, RP does not. That is evidence in F25 and F26 as we do no see a "U" shaped sloping down. The curve is mostly linear or "n" shaped. Increasing the number of random projections continued to have a large effect on decreasing reconstruction error. By running the RP multiple times, it can be observed that this pattern stayed true for these data sets. I select # original feature – 1 to minimize error.

F25



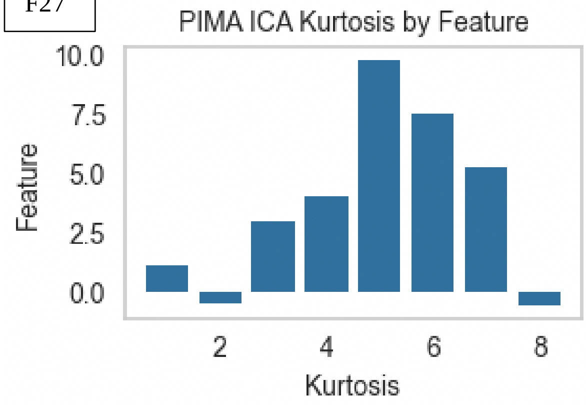
F26



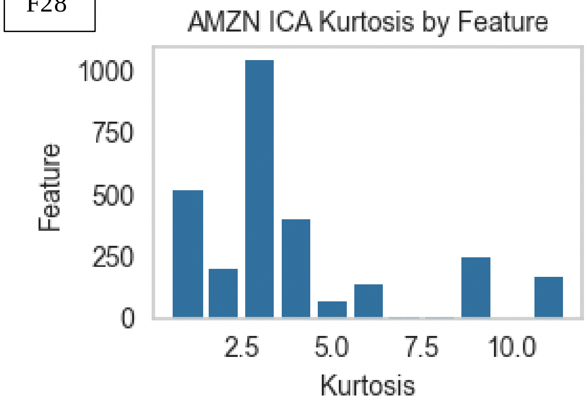
Dimensionality Reduction: ICA

ICA seeks to find the "true" causes of the observed data by recreating the data using independent components. I ran the ICA algorithm allowing for as many features as the existing data and then looked at the new components to find which ones to use. Specifically, I looked at which ones where the kurtosis is high (so there is a big peak instead of a fat slope). That indicates to me that it is not a normal distribution but more similar to a signal. F27 and F28 show Kurtosis (with the Fisher method, subtracting 3 from the answer so 0 in normal). We can see that some features seem to follow this desired pattern. I selected features with Kurtosis more than 1 for Pima and more than 10 for Amazon.

F27



F28



The data transformed by ICA will similarly project the same data into a new space but assumes that the dimensions are true signals that result in the original data (which will also have noise). Based on the high kurtosis of some of the data, I would expect good performance in prediction.

Dimensionality Reduction: RFE

RFE, unlike the other methods, uses the labels to select features. RFE uses a classifier (not a Neural Network) to decide feature importance. I was curious as to the impact of this classifier, so I ran the RFE with multiple classifiers. Here is show the difference in feature importance ranking using a Decision Tree (DT) and a Support Vector Regressor (SVR). As we demonstrated in Assignment 1, different classifiers have different strengths and weaknesses and will use the data differently. That is demonstrated in the table below for the Amazon Data (by feature order in dataset)

Amazon RFE Feature Selection Ranking By SVR and DT

5	8	2	7	6	3	1	1	9	1	4
6	1	1	5	1	7	8	4	9	2	3

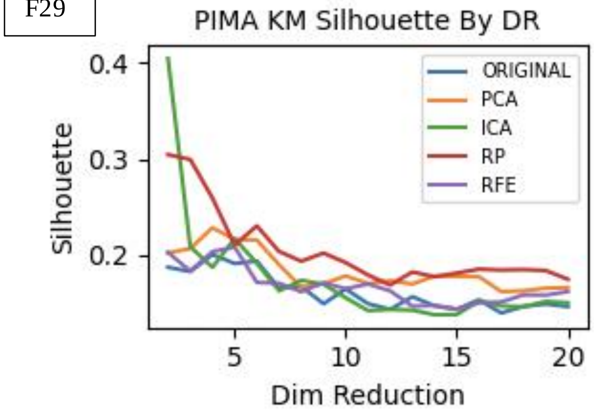
Clustering After Dimensionality Reduction

Looking at the Silhouette score for KMeans clusters using the original data vs using the dimensionality reduction data, we see generally that most reduction techniques did not improve the clusters for these datasets in terms of inter- and intra-distance.

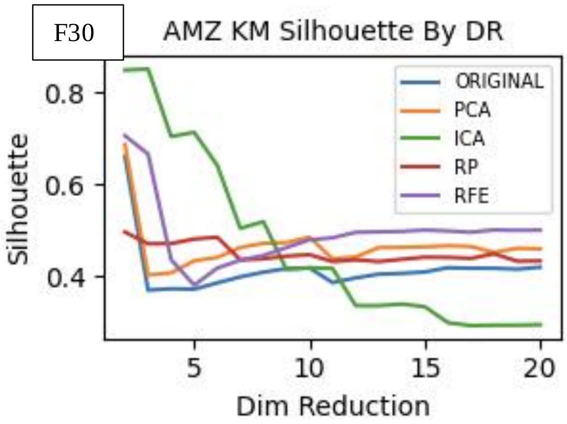
For Pima, we see a general "elbow" point (very fat elbow) between 5-10 clusters. We see the worst performance for RP, with PCA outperforming RP by a small margin We would expect this relative performance as PCA is more strategic about choosing its components. RFE stays about on par with the original. That makes sense as RFE will have the same data minus a couple of features that are deemed unimportant. ICA actually performs similarly to the original data for k>3, which would give reason to investigate this for additional insights.

For Amazon, we see the "optimal" k differ based on the dimensionality reduction used. We see k=3 for PCA and the original but k=5 for RPE, k=12 for ICA, and k=6 for RP. It is also very interesting to note that ICA achieves a very good score for k>11, beating out all others. This also raises good questions for investigation such as whether there are external true causes for the data. I also look at the LL of the components to create the data by the number of components for each of the reductions. For both datasets, PCA actually performs lowest which is surprising as we would expect the components to maintain most of the variance in the data. RP even outperforms PCA. RFE outperforms the original in the Pima data set, which makes me think that the dropped features were not very important for clustering. Most surprising though is that for both data sets, the ICA is able to choose EM components for which the data is likely to have occurred. The likelihood is actually much better than the other options. This is more supporting evidence of the need to investigate what independent components make up the data. It also could show a need to understand how independent components could help to maximize the likelihood (i.e. maybe there is something to how the ICA is created that makes the likelihood given of the data given the components higher). Multiple additional views did not give significant insights to add to this analysis, but I did find that the clusters did seem to change some (but kept their basic attributes that were discussed in the section above). When different k was recommended by the analysis, obviously those clusters are different, but the main difference seemed to derive from k instead of the reduction method used.

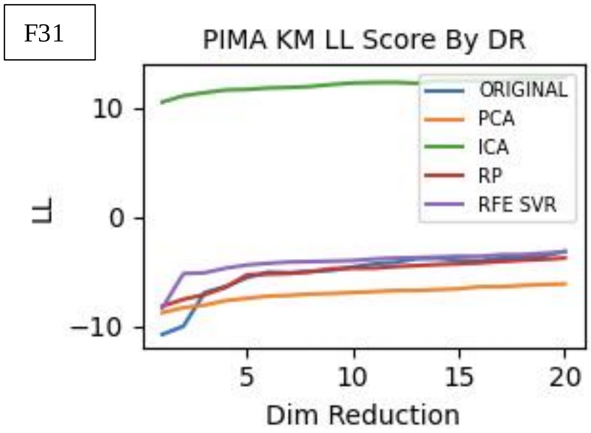
F29



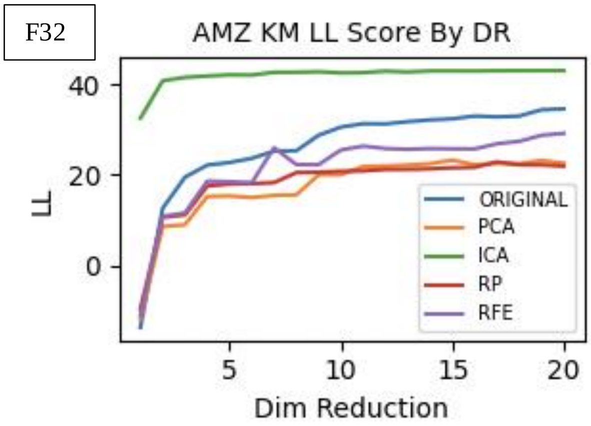
F30



F31



F32



Neural Network After Dimensionality Reduction

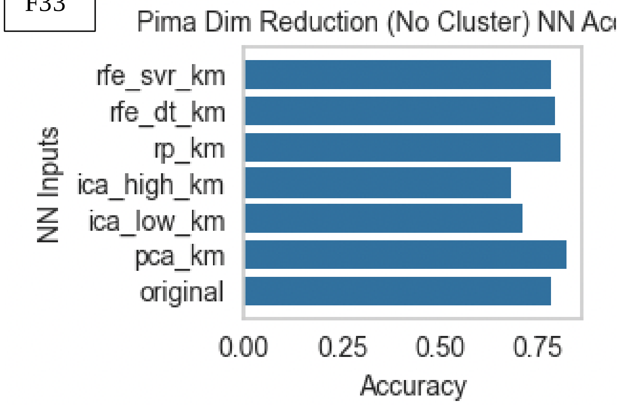
Selection: I after selecting the optimal k using the methodologies mentioned above, I then grid searched best parameters for a Neural Network for the data created with it. I used the same data set to train both which introduces bias, so it would have been better to split the data again.

I let the grid search explore different structures as well. I was surprised to find that many of the algorithm preferred a different structure than the original data (which was one hidden layer of 5 nodes). More surprising to me, though, was that the different reductions preferred different activation functions (only 'tanh' and 'relu' had come up from assignment one, but many preferred 'identity' and 'logistic').

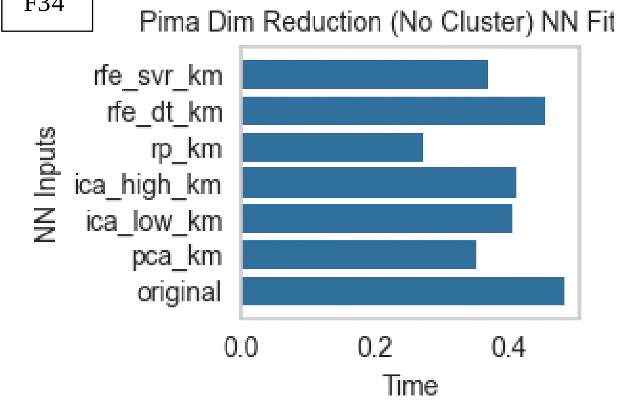
Comparing: I compared train and test accuracy and F1 scores along with time to fit for the Pima dataset. I first looked at the results using the reductions without clustering. I looked at two different RFE (using two different classifiers mentioned the the RFE section above), and I looked at two ICA with different Kurtosis cutoffs. "ica_high" chooses a higher kurtosis cutoff than the previous analysis (K=100 instead of k=10 for Amazon and K=3 instead of k=1 for Pima).

First off, F34 show more time to fit on average for the original features. We actually see the lowest time to fit for the RP which is actually surprising. I knew that the RP would do the reduction quickly (choosing a random), but this also shows that the result of the RP let the NN run more quickly. PCA gave the second most significant time decrease, which is a nice result to couple with the fact that PCA gave the highest accuracy (F33).

F33



F34



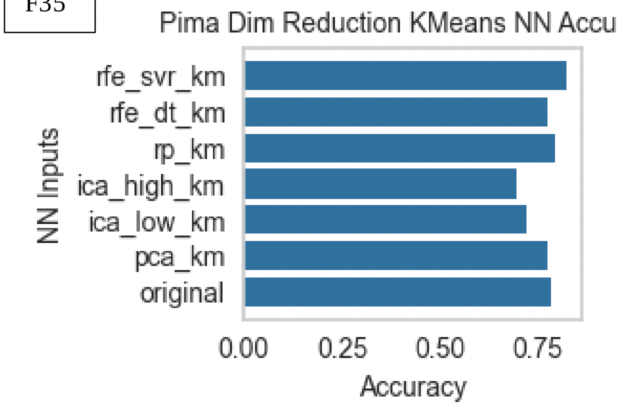
Neural Network On Clusters

I then looked at the results of dimensionality reduction plus clustering.

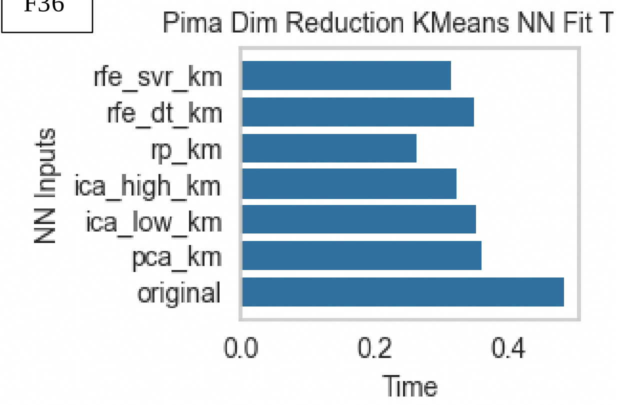
In the NN, the ICA actually gave the lowest accuracy no matter how I tuned the classifier (F35 and 37). This is very interesting give then the ICA gave the highest likelihood of the data given its projections for this same data (F31 and F32). Conversely, PCA had shown the lowest likelihood but performs fairly well. There was different performance for the RFE with different classifiers and ICA with different kurtosis thresholds. RFE with an SVR outperformed the original data for KM. Also, surprisingly the RP outperformed the original data (and PCA) for KM. PCA's poor performance may have to do with it not helping the clustering. No clustering combination outperformed the original data for EM, which was contrary to my hypothesis. I had assumed sense I could cluster and also share the probabilities with the NN that it would perform better in EM than in KM. We actually see the opposite across the board. We do see the EM on clustered data is much faster than KM (which is still faster than the original data) (F36 and F38).

I also ran all of the same data but kept the network structure the same (one hidden layer with 5 nodes) to find additional insights. The original data performed the same and most other options decreased slightly. Time was not affected significantly (structure did not change drastically).

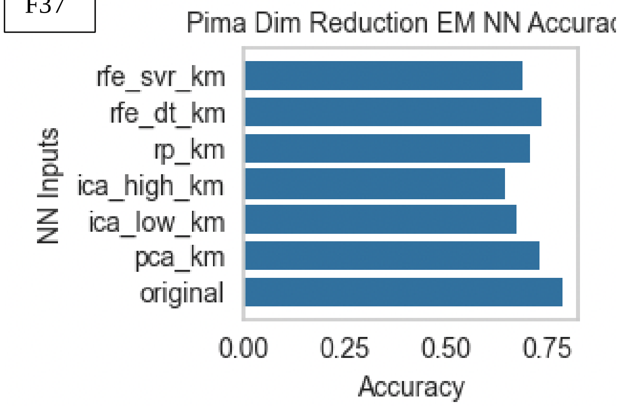
F35



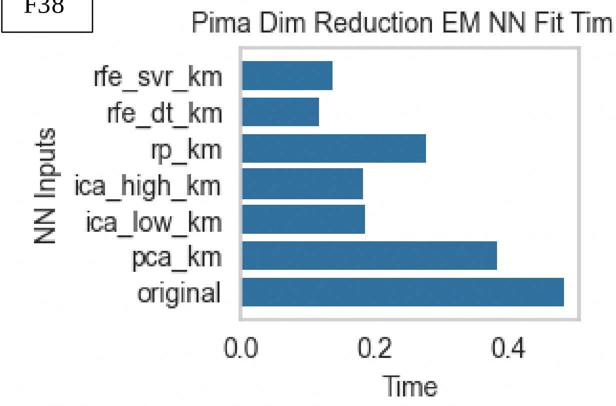
F36



F37



F38



Conclusion

Unsupervised learning in this assignment was different from the supervised learning from assignment 1 for a number of reason, but perhaps most notable was the need to create ways to measure “performance”. In labeled data we know what we should get, and many smart people have created metrics to compare them (e.g. F1 score). In unsupervised learning, many smart people have also created ways to compare performance (e.g. silhouette score), but “better” is much more vbigague and depends much more on the situation and domain knowledge. Overall unsupervised learning seems very useful for exploring and finding new insights.

Work Cited

1. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011. https://scikit-learn.org/stable/about.html
2. https://www.scikit-yb.org/en/latest/about.html#citing-yellowbrick