Unsupervised Learning and Dimensionality Reduction

Stephen Shepherd | sshepherd35@gatech.edu | 03/24/2022

Introduction

In this paper I apply two Clustering and four Dimensionality Reduction approaches to two labeled datasets and assess the results in terms of their usefulness in data mining and classification.

Datasets and Preprocessing

I used the two datasets I explored in Assignment 1. The Heart Disease dataset consists of 27 features of behavioral health indicators like high cholesterol and fitness activity, with a label for previous diagnosis of heart disease or not. The Digits dataset consists of 64 features representing pixel intensity for images of the handwritten digits labels 0-9. Categorical features (only present in Heart) were one-hot encoded. All features were standard-scaled (fit on the Training Set and applied to both Training and Test sets). Heart contains ~38k Training samples that are 50/50 "balanced" by label, and ~50k holdout Test samples that remain imbalanced at 90/10 no/yes. Digits contains ~1.4k Training samples and 360 holdout Test samples.

K-Means ("KM") Clustering

For K-Means, I ran a number of trials over a range of *k* values and chose *k* based on the best Silhouette Score metric. This metric "compares the mean intra-cluster distance (*a*) and the mean nearest-cluster distance (*b*) for each sample. The Silhouette Coefficient for a sample is (*b* - *a*) / max(*a*, *b*)."[1] While I also tracked a squared distance to the nearest centroid metric, I found that this value can be meaninglessly improved by "overfitting" with many clusters. Highest silhouette scores were conveniently located near the "elbows" of the distortion, although a distinct elbow for the Digits dataset was more elusive compared to the one for the Heart dataset.

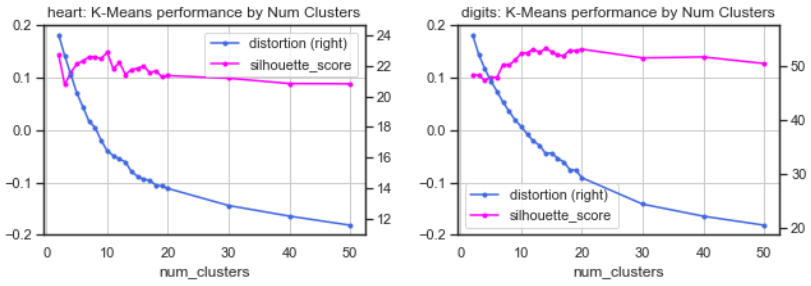


***Figure 1**: K-Means performance by number of clusters.*

Interestingly, the Heart data cluster Silhouette Score was nearly identical at *k* of 2 and 10. While the Heart data has 2 true labels, I found from Assignment 1 that the Heart dataset is very noisy so I elected to model 10 clusters instead of 2 to see if they isolate some of that noise given the similar Silhouette Score but lower distance metric at 10. Similarly, we see a correlation between performance at *k* and the # of labels for the Digits data: the Silhouette Score spikes as *k* reaches 10 with the maximum at 14, which I ended up using. Intuitively, the largest Heart cluster (6) correlates roughly with the target disease label, although

[1] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

imperfectly. For Digits, the digits 0, 3, 4 ,5, 6, 7 and 9 map strongly to one of the clusters, whereas 1, 2 and 8 are distributed across a handful of clusters: this is intuitive as digits 1 and 8 were among the toughest to classify for a Boosted Trees learner in Assignment 1. The digit 1 is noisy as sometimes it's drawn with a flag, and the digit 8 has many pixels that overlap with many other digits and a circular shape.

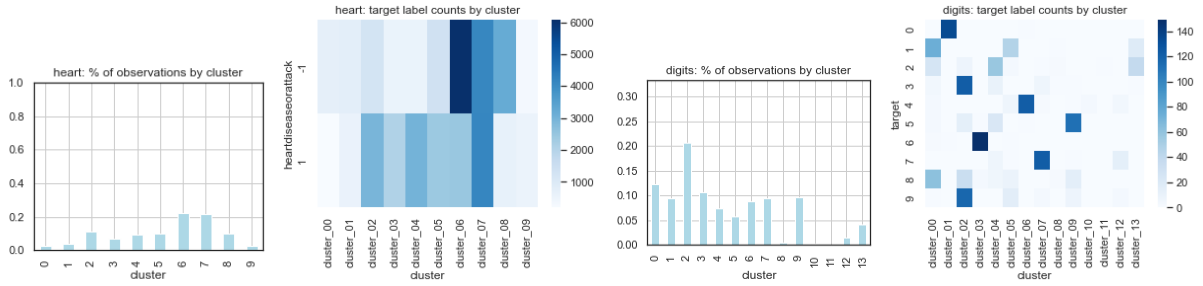


***Figure 2**: K-Means cluster observation counts and cluster correlations with the target labels.*

Cluster 6 for Heart, which correlates with the no disease label, correlates positively with better general health, no diabetes and higher income. Cluster 1 for Digits, which maps almost exclusively to the digit 0, resembles a 0 in its average pixel values. Clusters 0 and 5 both map to many digit 1 labels, but Cluster 0 appears to capture 1's drawn with no flag while 5 appears to capture 1's drawn with the flag. It's exciting to see K-Means identify this distinction for different styles of representing the same digit. The Digits clusters 8, 10, 11 and 12 are small and contain high intensity for pixels in the 4 corners and likely map to digits that were drawn off-center. Given their low prevalence and outlier nature, I might consider running a learning preprocessing step on Digits that attempts to "center" each digit to avoid K-Means detecting the clusters we see mapping to the corners.

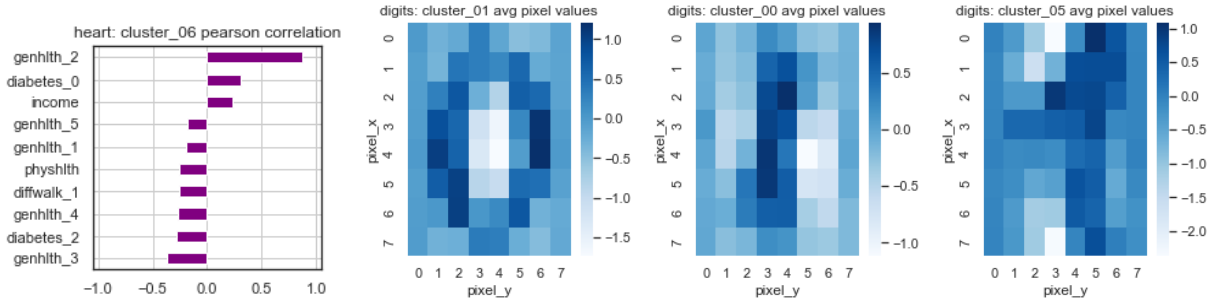


***Figure 3**: K-Means cluster correlations with features.*

Expectation Maximization ("EM") Clustering

Using EM, Silhouette Scores (which proved less helpful than BIC) for both datasets were maximal at 2 clusters. This is intuitive for the Heart dataset which has a binary disease label, but is unintuitive for Digits with the 10 digit labels. Examining the observations predicted for each of the two clusters, we see one dominant cluster containing > 97% of the observations for both of the datasets. This does not happen with K-Means using $k = 2$. For Digits, the small cluster contains just one observation for the digit 2 with high pixel intensity in the lower left corner. EM appears to be modeling outlier samples in both datasets when asked to predict just 2 components/clusters. These outliers would come from a different distribution than the majority of the samples. While K-Means had a more balanced distribution, it likely merged these outliers into one of the two clusters, and they probably negatively influenced the centroid of the cluster(s) to which they were mapped by skewing the mean.

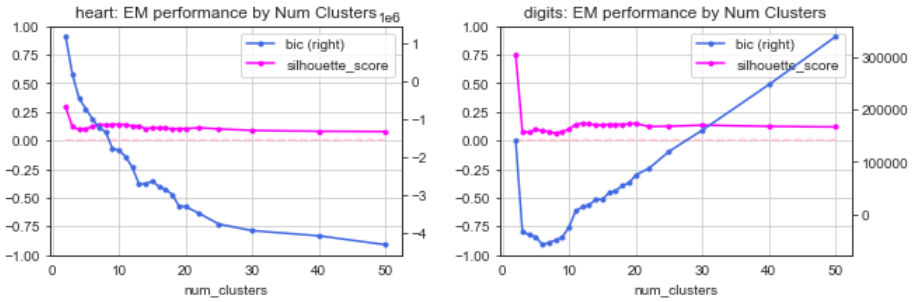


***Figure 4**: EM performance by number of clusters.*

Algorithm	Data	Cluster 0 Observations	Cluster 1 Observations
K-Means	Heart	64%	36%
	Digits	62%	38%
EM	Heart	98%	2%
	Digits	100%	0%

***Figure 5**: % of observations by cluster, comparing K-Means vs. EM modeling just 2 clusters.*

Using BIC to evaluate the clusters, we see a minimum BIC reached for Digits at 6 clusters (which I used), increasing sharply after 10 which is intuitive as this happens to be the # of true digit labels. BIC for Heart decreased somewhat monotonically and elbows at 14 (which I used).

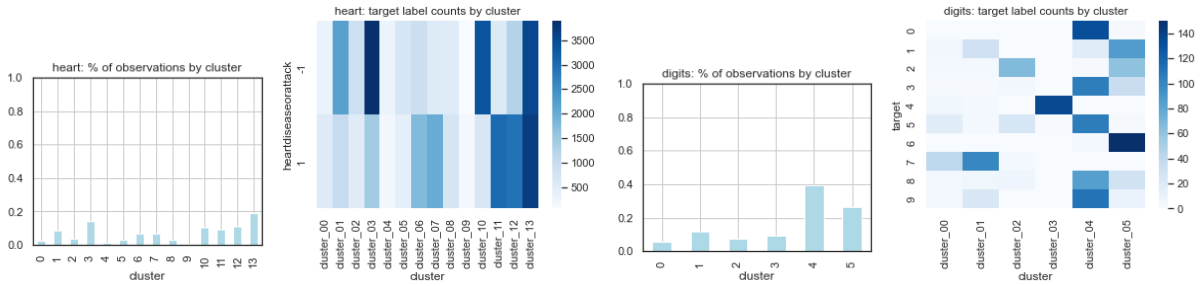


***Figure 6**: EM cluster observation counts and cluster correlations with the target labels.*

For Heart, we see cluster 3 correlate with the no disease label, and healthier fitness indicators and higher income, similar to K-Means cluster 6. We also see cluster 13, which has an even mixture of disease labels, correlate with more intermediate health indicator values like General Health = 3 of 5. This is similar to the K-Means cluster 7. This cluster appears to map to the noisier samples in which the features are less helpful and the labels are harder to delineate, which is concerning given the high % of samples that map to it (nearly 20%). This validates my Assignment 1 findings that show the Heart data is noisy, containing lots of positive and negative label collisions for observations with similar feature values. It's nice to see EM capture this set of confusing observations. For Digits, the BIC for EM suggested using less clusters than the Silhouette Score did for K-Means, so there is more cross-pollination between clusters and labels here. Cluster 4 (capturing ~40% of samples) was particularly interesting as it maps to digits that have curves at the top and bottom, such as 0 and 9. Cluster 5 maps to the digits 6 and 1 which can contain straight shapes vertically to the left and also presence in the lower right corner. These examples are shown in *Figure 7*.

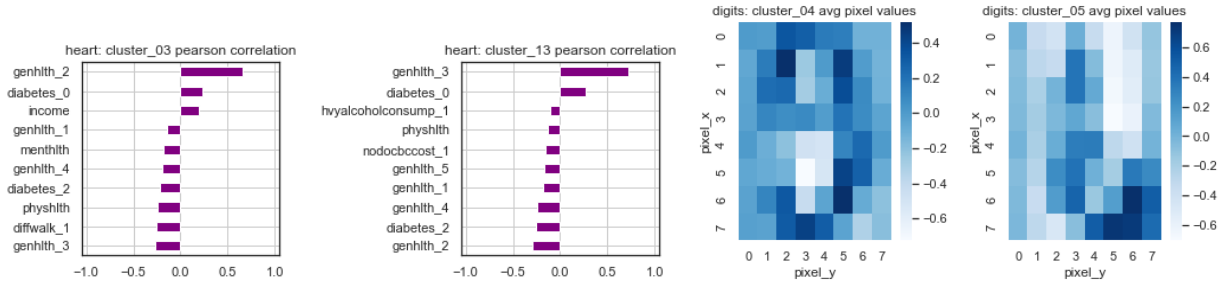


***Figure 7**: K-Means cluster correlations with features.*

Principal Components Analysis ("PCA")

I then ran PCA on both datasets, using the first *k* components that totaled 90% of explained variance. This was 20 for Heart (out of 27 original features) and 31 for Digits (out of 64 original features). The Heart data required more components relative to the total # of features to reach 90%. Both datasets reached ~100% of variance explained with a $k <$ # of original features, indicating some features can be nearly fully represented in the lower dimensional projections.

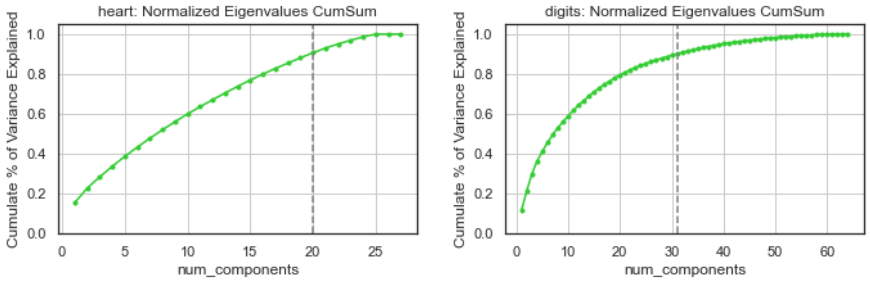
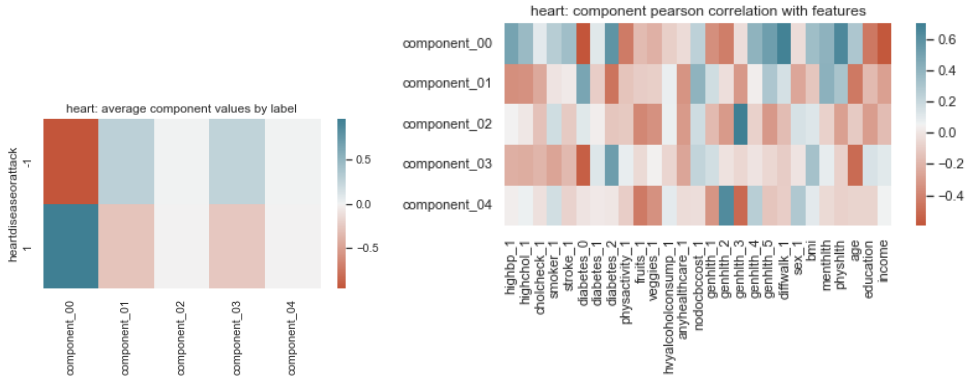


***Figure 8**: Normalized Eigenvalues cumulative sum by # of Components.*

For Heart, the first component is correlated with the disease label and the intuitive risk factors like older age, diabetes, high blood pressure and lower income. This one direction seems to capture much of the variation across these correlated features and project them into one component. For Digits, the components seem to decompose the pixel intensity into correlated "regions": component 1 has positive correlation with the upper right corner and negative correlation with the lower right corner, for example (see *Figure 9)*. It's easy to see why it's positively correlated with the digit 7 and negatively correlated with the digits 6 and 0.



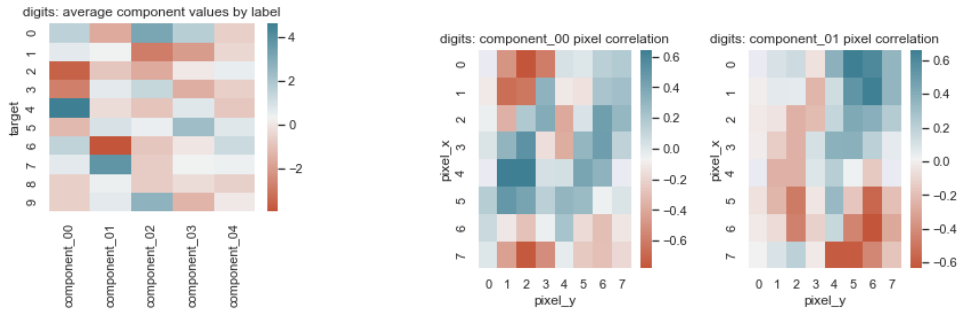


***Figure 9**: Component interactions with features and labels for both datasets for the first 5 components.*

Independent Components Analysis ("ICA")

I explored Independent Components Analysis as an alternative to PCA. This seeks to identify source components and separate them, rather than tending to combine things together. I used the average absolute value of the Kurtosis of the components to select the # of components. Negentropy is another metric for this that I didn't have time to explore. Via "elbow," I used a *k* of 16 for Heart and 22 for Digits.

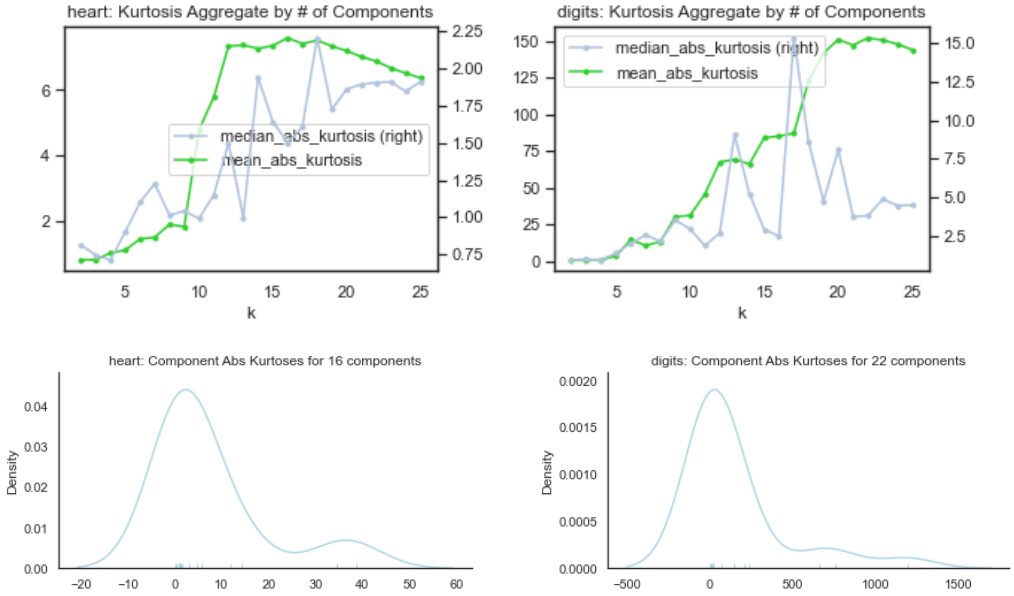


***Figure 10**: Average and Median absolute value of Kurtosis of the generated components by # of components modeled, and a distribution of the absolute value of the Kurtosis of the components.*

Notice in *Figure 10* the lower Kurtosis extremes for the Heart dataset relative to Digits. Some of the Digits components showed extremely high Kurtosis, and I found that these corresponded to outliers in the Digits dataset. For example, Component 17 for Digits, which has a Kurtosis > 1k, contains an unusually high value for an example of the digit 2 that contains high intensity in the lower leftmost pixel which is blank in most of the other examples. Because the highest kurtosis components mapped to outliers on Digits, I found that the mid-range Kurtosis components represented more generalizable patterns. This can be seen in *Figure 11* where the intermediate components (sorted by Kurtosis) contain the strongest correlations to the actual Digit labels. These components appeared somewhat similar to those generated by PCA because they represent regions such as corners, the right angle at the top left of a 5 (see Digit Component 1 in *Figure 11*), or the hole in a 6. For Heart, the kurtosis of the most distinct components is

more moderate than digits, and we see these map to more generalizable patterns in the features. Components 8 and 10 correlate with Diabetes_1 and CholesterolCheck_1 respectively. That being said, the 6th most extreme component is where the components start correlating strongly with the disease label.

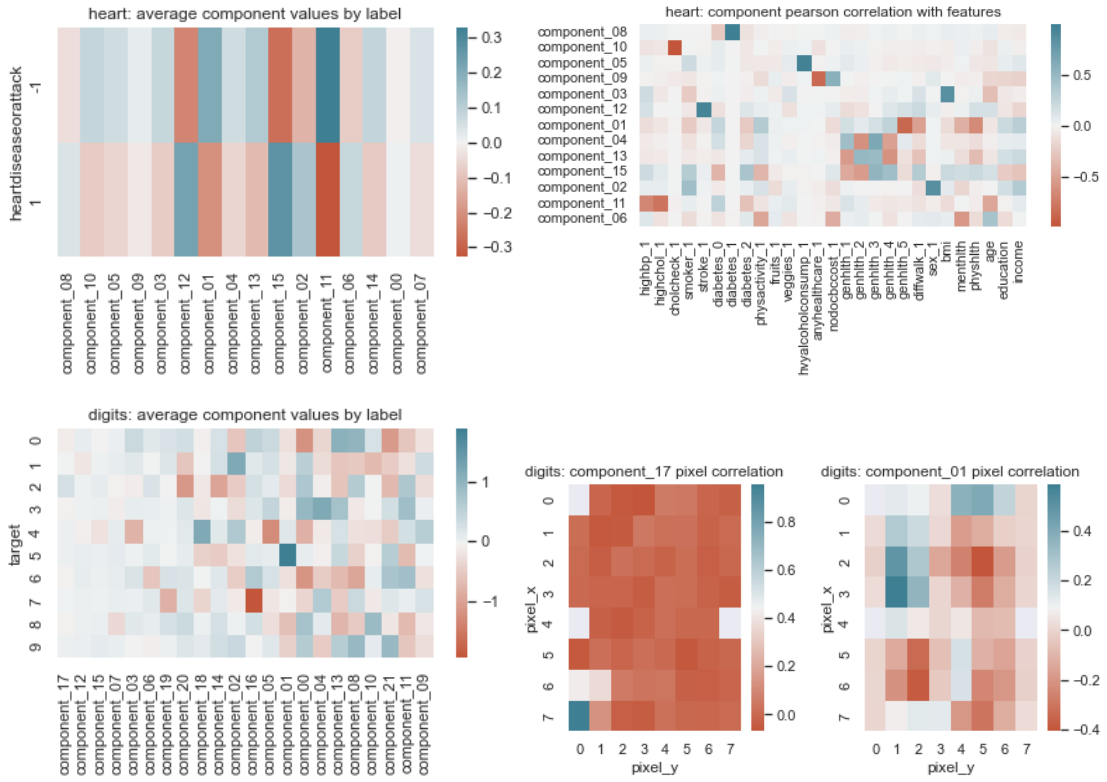


***Figure 11**: Component interactions with features and labels for both datasets. Components are sorted from highest absolute value Kurtosis to lowest where multiple are plotted (eg. 08 is highest for Heart, 17 for Digits).*

Randomized Components Analysis ("RCA")

I explored Randomized Components Analysis as a less structured alternative to PCA and ICA. I chose *k* at the number of components that produces a R-Square of 75%. I did this to decompose the dataset into a subset of the information while still keeping a healthy majority of the information. This resulted in a *k* of 21 (27 original features) for the Heart data and 49 (64 original features) for the Digits data.

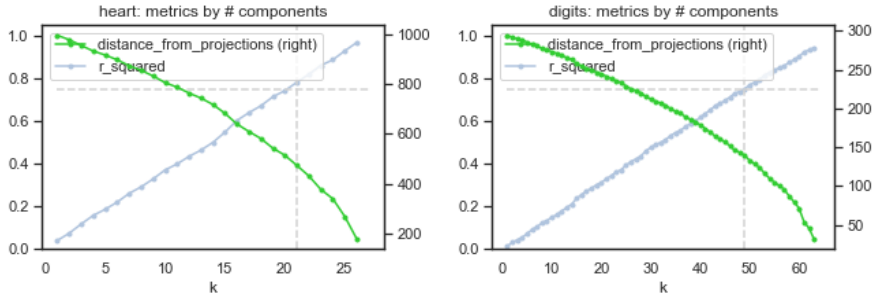


***Figure 12**: Distance from projections and R-Square by # of components.*

The components generated by RCA are not necessarily orthogonal or independent like those generated by PCA and ICA. Accordingly, multiple components show similar correlations to the target labels and the original features. For example, Components 1 and 12 for Digits correlate with the digit 2 but focus on different regions of the digit, as demonstrated in *Figure 13*. Combined together, these two components could model the digit 2 fairly well. On Heart, multiple components correlate with the disease label, but component 15 stumbled upon this moreso than others, correlating negatively with heart disease. It correlates positively with good exercise and diet features. Component 6 similarly correlates negatively with heart disease and concentrates positively with the better general health features.

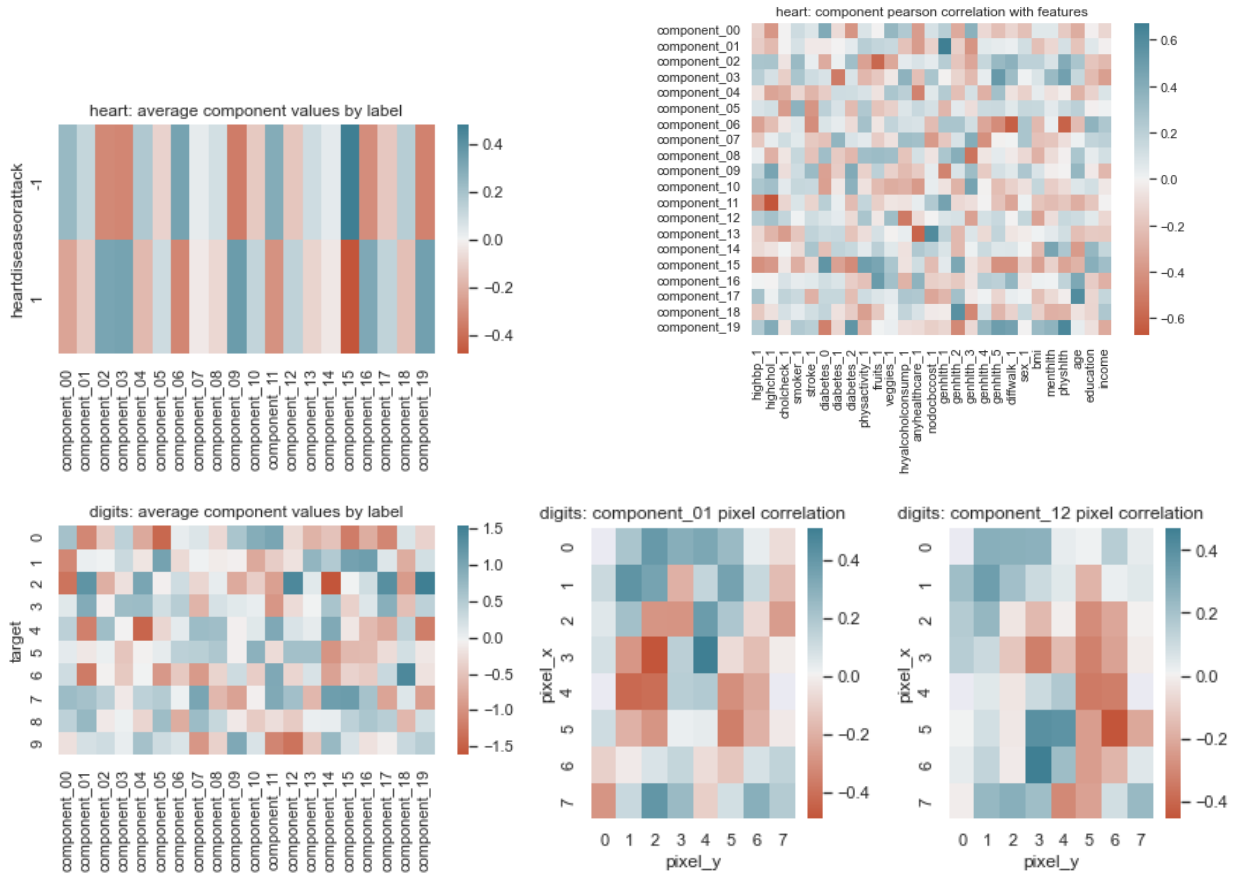


***Figure 13**: Component interactions with features and labels for both datasets (only 20 components for brevity).*

This more "shot in the dark" style approach at randomly guessing projections does seem to produce some decent correlations with the true labels and identifies some interesting patterns in the features. However, compared to PCA, more components are required to capture a decent share of the variation in the original data. I'm interested to see how this approach compares to the others when used as the features for Supervised Learning.

Feature Selection via Filtering

I explored a Filtering feature selection process that combined an AdaBoost classifier with a stepwise Backward Search approach. I chose the tree-based classifier (as Dr. Isbell suggested in the lecture) as it selects features based on Information Gain, which is a simple way to discover relevant and useful features that may interact non-linearly with the target labels. The classifier was able to retain decent cross-validation accuracy on both datasets with less than 50% of the original features: 13 of 27 for Heart

and 17 of 64 for Digits. Quantities below these values "elbowed" off into poor accuracy, especially for Digits. In the Heart dataset, many of the features are correlated with each other so the elbow is less pronounced: accuracy varies less even as the # of features used diminishes, as some features can represent similar information from other correlated features.

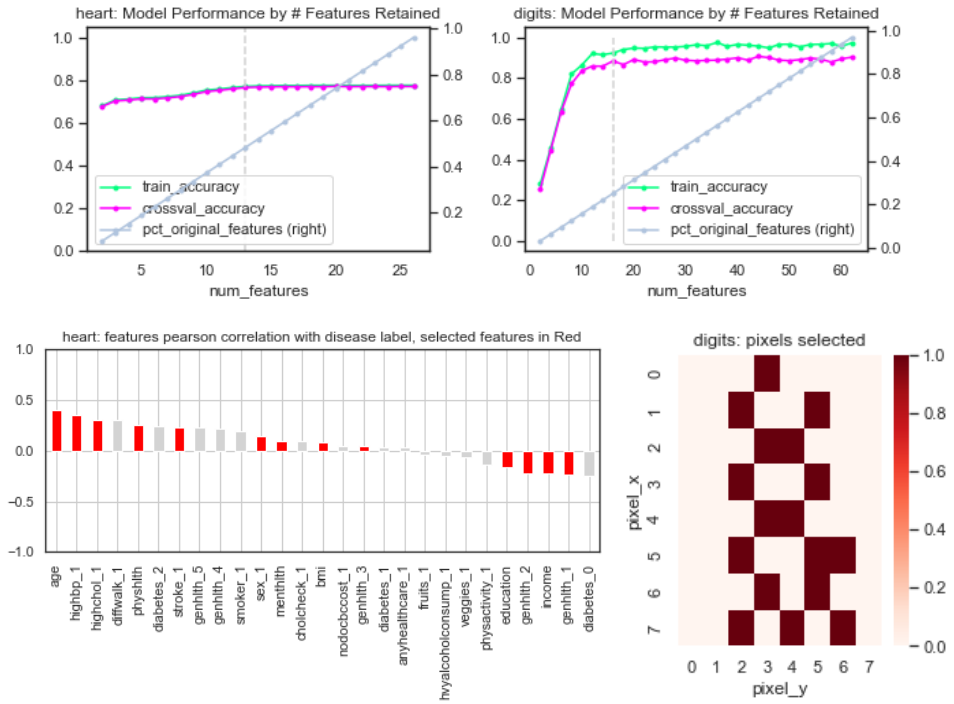


***Figure 14**: Filtering Feature Selection results. Top: accuracy by # features retained. Bottom: highlighting the selected features.*

For Heart, the selected features tend to have higher linear correlation with the target label, but some with low correlation were picked probably as helpful tie-breakers vs. other more correlated features that were probably correlated with features already chosen. The pixels chosen for Digits are concentrated in the middle and bottom of the drawing space, avoiding the top edge and the 4 corners with the exception of the lower right corner. I'm excited to see what kind of results that Clustering and the Neural Network produce with these subsets of features.

Clustering on the Dimensionality-Reduced Data

I retried clustering for each dataset using the dimensionality-reduced data (from PCA, ICA, RCA, Filtering Feature Selection). For K-Means, the best Silhouette Score achieved using the original features was ~0.15 for both Heart and Digits. ICA and Feature Selection reduced data were able to exceed this baseline for Heart, and everything except PCA was able to beat this for Digits. It appears that dimensionality reduction was able to simplify the clustering problem for K-Means, perhaps combining some of the less distinct features into a more compact space. Interestingly, ICA produced an outstanding Silhouette Score on Heart of .4 at 2 clusters, and PCA and ICA produced even higher scores than this at 2 clusters for Digits. At 2 clusters for both PCA and ICA, Cluster 1 contains < 10 samples and appears to be capturing the outlier samples with high intensity in uncommonly drawn pixels. These small outlier clusters seem to be captured well by these reduction techniques (as discussed in those previous sections) and is skewing the clustering. In the class lectures, it's mentioned that dimensionality reduction can

model overarching factors like total brightness or defining edges - samples that are outliers for these factors should probably be identified and removed before applying Clustering.

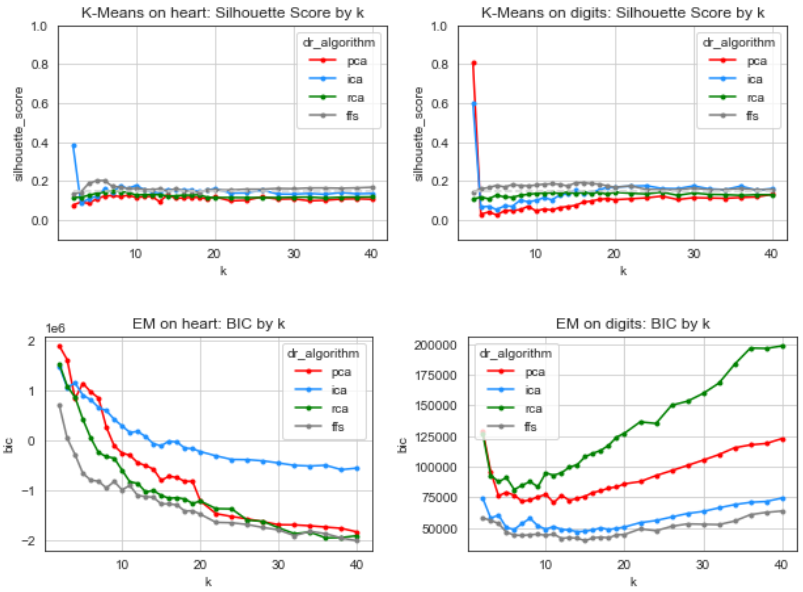


***Figure 15**: K-Means and EM performance on the dimensionality-reduced data.*

EM using the dimensionality reduced data was not able to reach the lower BIC values seen with the original features. The Heart values appear nearly monotonically decreasing alongside *k* as was the case with the original features, while the Digits values do "elbow" at lower k's before increasing, although they elbow at higher *k*'s than was the case for the original features. I was surprised to see some of the highest Silhouette scores and some of the lowest BIC scores come from the Feature Selection data: perhaps representing most of the important original information in it's original form can have benefits for Clustering, as opposed to to running Clustering on projected data. For both K-Means and EM, the PCA reduced data was more likely to lead to the largest cluster modeling a region (a curve in the lower right corner) instead of something resembling an actual digit. This suggests that PCA + Clustering could be used to identify helpful overarching patterns instead of more specific trends that might correspond with true labels.

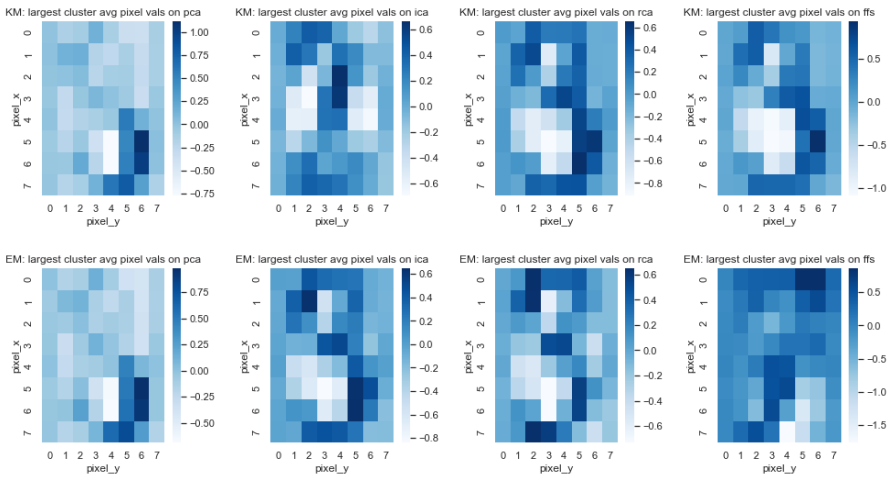


***Figure 16**: K-Means and EM avg. Digits pixel values for the largest cluster produced at k=10.*

Neural Network Performance on Reduced Digits Data

I trained my Neural Network from Assignment 1 on the Clustered and Dimensionality Reduced data using the *k* values arrived at in the sections above. For K-Means, I used a column for each cluster containing the distance to cluster centers and for EM I similarly used the predicted probabilities for each of the clusters. For PCA, ICA and RCA I used a column for each projection. I standard-scaled all of the resulting (reduced) data, fitting the scaler on the (reduced) training set. Overall, the reduced data was able to be learned in a way that allowed the model to achieve similar accuracy to the original data using a fraction of the # of dimensions. The exceptions were Expectation Maximization and the Filtering Feature Selection approach. For EM, the small # of clusters seems to limit the performance, although I tried EM with 12 components and it was only able to achieve ~74% cross-validation accuracy. With 6 components, Cluster 4 (the "curvy" one shown in *Figure 7* that maps to multiple digit labels) seems to be confusing the model by offering one large component that correlates with multiple labels. The NN incorrectly predicts many 0, 3, 8 and 9 labels as 5's. While I can see that this cluster offers insight into an interesting generative distribution that is common to multiple digits in handwriting, it offers a confusing signal when the model is trying to delineate. For the Filtering Feature Selection, it appears that dropping some of the pixels reduces generalization accuracy while the Projection approaches like ICA are able to capture the information from the original features in a lower dimensional space that doesn't hamper generalization.

dataset	algorithm	k	train_accuracy	crossval_accuracy	test_accuracy	train_seconds
digits	Original Features	64	1.00	0.97	0.97	0.14
	K-Means	14	0.96	0.94	0.94	0.29
	Expectation Maximization	6	0.42	0.44	0.38	0.06
	Principal Components	31	1.00	0.96	0.97	0.13
	Independent Components	22	1.00	0.96	0.96	0.14
	Random Components	49	1.00	0.95	0.96	0.12
	Filtering Feature Selection	17	1.00	0.94	0.93	0.22

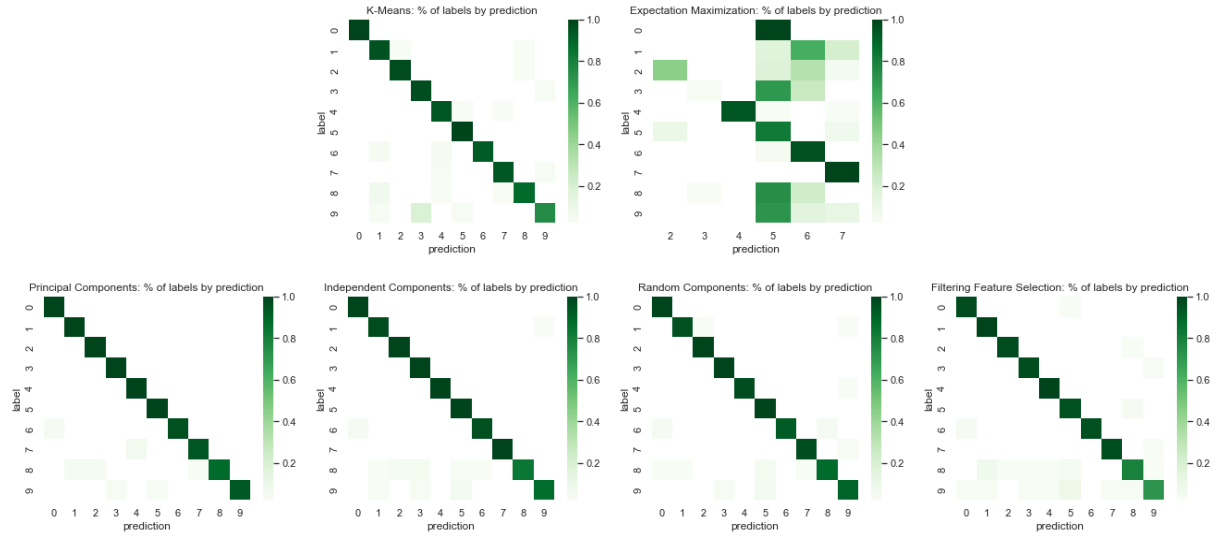


***Figure 17**: Neural Network classification performance on Digits by the Dimensionality Reduction result data used for training.*