Ben Fuqua

Assignment #3

Unsupervised Learning & Dimensionality Reduction

For this assignment I am using a breast cancer dataset and an image classification dataset. The breast cancer dataset comes from Sklearn and consists of various measurements and statistics about breast cancer. This dataset has purely continuous variables with a binary target. The dataset has a slight class-imbalance with 212 in the negative and 357 in the positive for a total of 569 samples. With this imbalance, a classification model would lean slightly towards predicting breast cancer which increases the risk of a type I error but with this being about cancer it is better to error on the side of caution. The second dataset is an image classification dataset, with ten targets and the features are the RGB codes for each pixel which are then standardized by dividing by 255 forming a ratio between 0 & 1. I narrowed it down to only two targets for simplicity's sake. These classes are perfectly balanced with five thousand records each, this will also prevent any sort of bias when the data is trained in the model, unlike the breast cancer data. We see multiple points of interest in the datasets that can and will affect the way the model performs because one being balanced, and one not; one with lots of samples and one without.

For part 1 of this analysis, we be doing the following on both datasets, we will first explore how our 2 clustering methods perform (K means, Expectation Maximization), then our 4 dimensionality reduction algorithms(Principal Component Analysis, Independent Component Analysis, Random Component Analysis and Linear Discriminant Analysis), then each clustering method with each reduction for a total of 8 tests. So, 8 + 4 + 2 = 14 * 2 = 28 different tests will be run. For part 2 we will be exploring how these algorithms perform on our neural network from assignment one. Later, I will explain the various choices I made for the parameters of these algorithms.

Part 1: Exploration

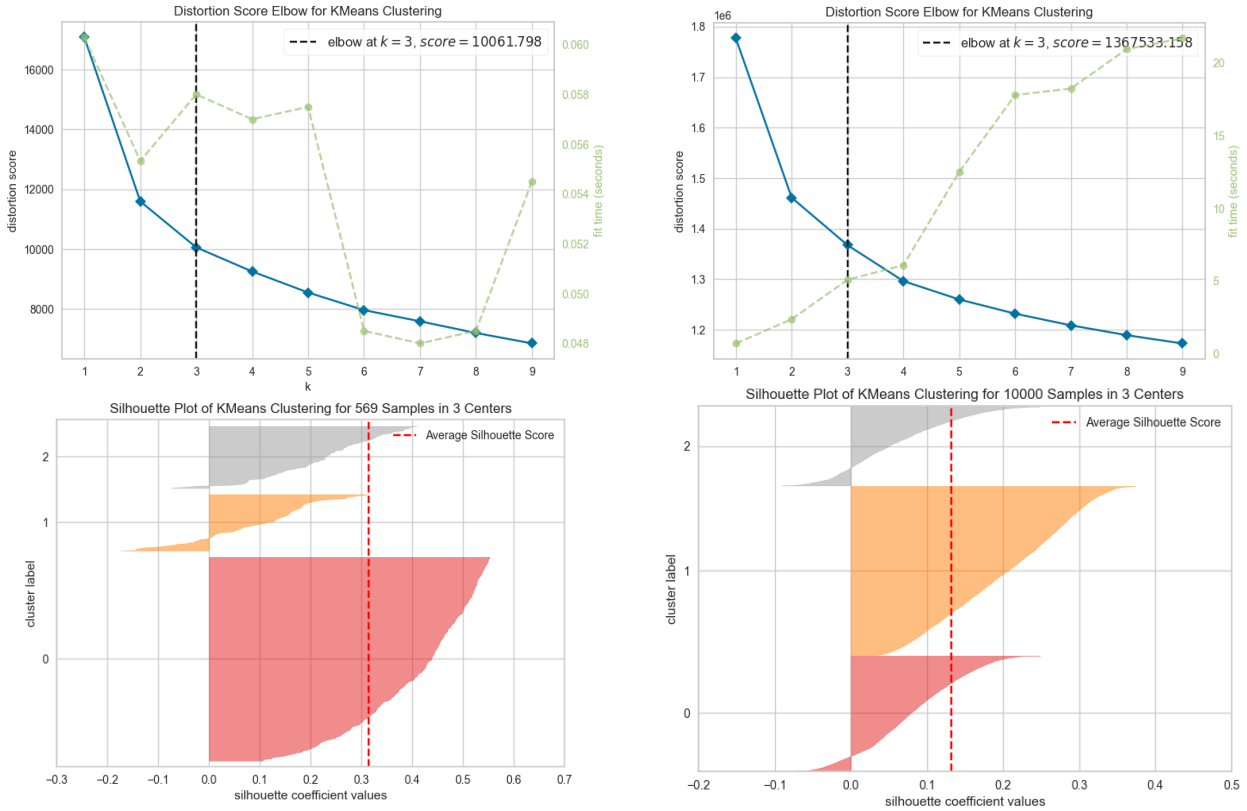
Cluster Algorithms

There are two clustering algorithms we will explore for this assignment; K means Clustering and Expectation Maximization or KM and EM respectively. The KM algorithm is remarkably similar to that of K-Nearest Neighbors in the sense that it forms clusters in n dimensions of space and giving them a classification based on proximity to the center of the clusters. Two metrics for this distance are Euclidean and Manhattan, same as in the KNN model. Because the KM model is an unsupervised model, we do not have the ability to predefine clusters, so KM must figure it out. It starts by picking k centers at random anywhere on the plane then each point is assigned to a cluster based on proximity and the centers are

recomputed and then shifted towards the center of the points assigned to them. Repeat this process until the algorithm converges.

EM utilizes something called ‘soft clustering’ which is nothing more than the KM algorithm, but one point could be assigned to more than one cluster if it looks like it could belong to both clusters, so it generates a probability score. These scores are called ‘hidden variables’ and they are used to help determine which cluster the point should belong to, think of it as a way of breaking ties. The process of EM is to pick two points at random and then assign all points to a cluster while creating a band that is not strongly pointed to one cluster or the other. While this is one of the pros of the EM algorithm, the downside is that points that clearly belong to one cluster still have a non-zero probability of belonging to the other cluster. This is because the gaussian distribution can go on for infinity, so it is possible to have a point way out in the tails, even though the probability is .000004%. Now that we have explored these algorithms, let us put them to practice.

KM & EM



Above we have an elbow and silhouette plot. The elbow plot uses distortion as its metric which is simply the average of the squared distances from the point to the assigned cluster, so the tighter the cluster the better. The silhouette score takes this into account, plus how far apart each cluster is away from the other clusters, so tight clusters far away from each other is

the best for this metric. The elbow plot helps to determine the optimal k values, in our case three for both datasets, and then with the k value you can generate the silhouette plot. Based on the plots, we can see the breast data (on the left) is able to form tighter clusters than the image data (one the right) and is able to get a better silhouette score as well. This tells us the breast data has clusters that are tight and farther apart from one another. This data was trained on all the available data so I cannot visualize it, yet. That comes in the dimension reduction section. It is worthwhile to note though that when I predicted the cluster based on KM or EM I would get anywhere between an 80%-85% accuracy rate.

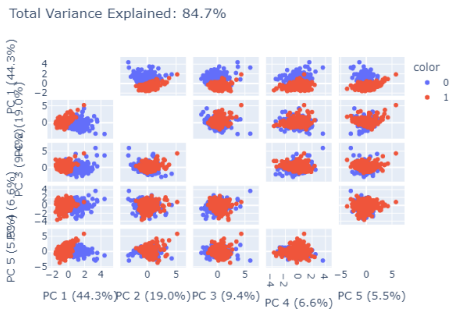
Dimensionality Reduction Algorithms

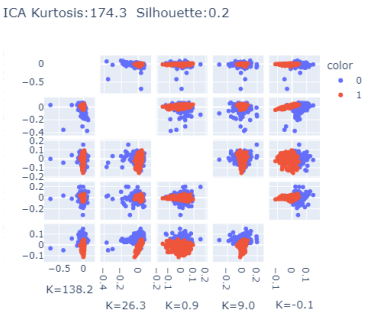
Principal Component Analysis (PCA), Independent Component Analysis (ICA), Randomized Component Analysis (RCA) and Linear Discriminant Analysis (LDA) are the four items explored. PCA focuses on maximizing the variance of the data by creating a line of best fit through the data. Once it finds that line, it then finds a line that is orthogonal to it. The way we judge the importance of each component is by looking at the total variance explained, or the eigen value associated with an eigen vector. Starting with the principal component, the eigen values monotonically non-increase and we can see how quickly we are able to capture the information within the data. With PCA there is no loss of information but is collapsed into less dimensions.

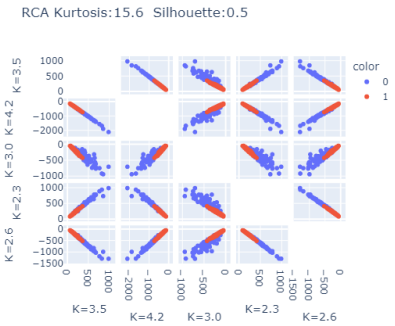
ICA strives to increase the independence of each variable, but make the corresponding original variable be high dependence. Through a linear transformation it strives to make the mutual information between all variables in the new space be 0 and the mutual information from space 1 -> space 2 be as high as possible. ICA as well as RCA and LDA are measured by something called kurtosis (NOTE: I mean excess Kurtosis and not Kurtosis). Kurtosis is the measure of the sharpness of the peak of a frequency distribution. There are 3 types of kurtosis: Leptokurtic (>0), Mesokurtic (0), Platykurtic (<0). Leptokurtic has a high peak and small base, Platykurtic has a short peak and wide base, and Mesokurtic falls in the middle of these. For reference, a normal distribution has a kurtosis of 0 and kurtosis ranges from negative infinity to positive infinity.

RCA is exactly what it sounds like, random components projected into your space. This works well in practice; this is due to the curse of dimensionality. In general, RCA does not get you down to the lowest number of dimensions, but it still gets you low enough to where you do not have to worry about the curse of dimensionality. RCA is still able to notice correlations between variables, store that into a single project and reduce your data.

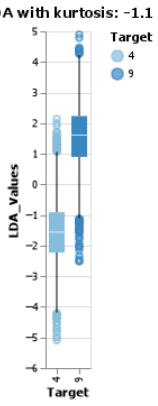
LDA is remarkably similar to supervised learning in the sense that it finds a project that discriminates based on the label. Compare it to an SVM, LDA tries to find linear separators between two different clumps of points, or in an SVM we would call that a support vector. Where the other 3 methods don't care about the label, LDA tries to use it to find the best projection that separates them the best.

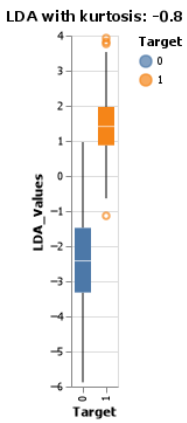


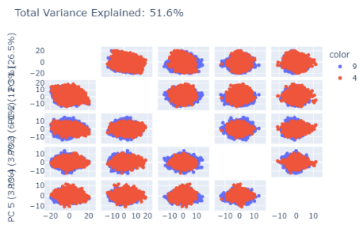


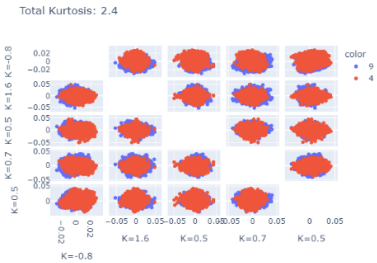


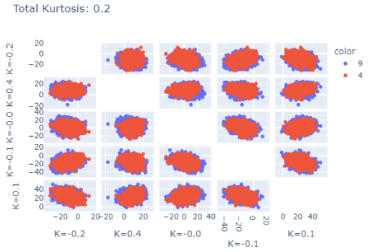
Above is data from the cancer dataset with the true datapoints being labeled in blue and red colors, from left to right we have PCA, ICA, and RCA with LDA to the left. Visually we PCA, RCA and ICA can create defined clusters with their components. Due to the limitations of LDA needing min (samples, num_features -1) we could only reduce it down to 1D, but none the less we were still able to see the division of clusters in a boxplot. Due to RCA making random projections we do not see very high variance or even independence on this data, which is why we see the more pronounced lines and some flaring on one side or the other. Below we have the same test but on the image data with LDA on the right. Unlike the cancer data we are not able to form distinct clusters. This helps us to understand kurtosis better as the data is more group together, it approaches a normal distribution, but as they become more separate, they form their own peaks which can either have fat or skinny tails. Our LDA model seems to have done the best as there appears to be around a 25%-30% overlap between the clusters.



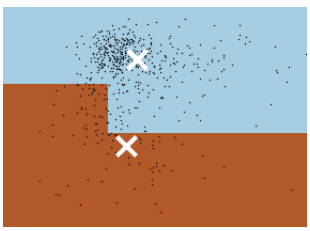


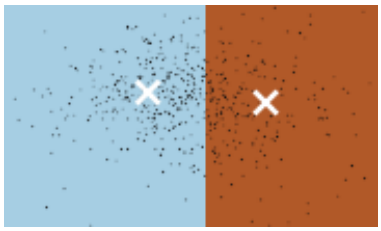


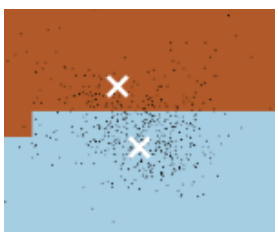


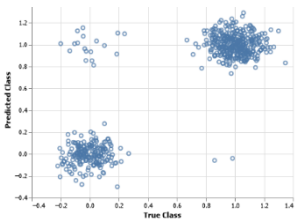


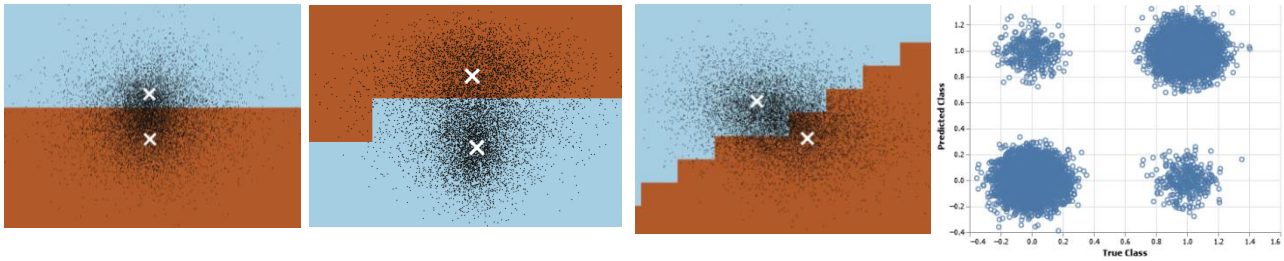
Cross Examination & KM







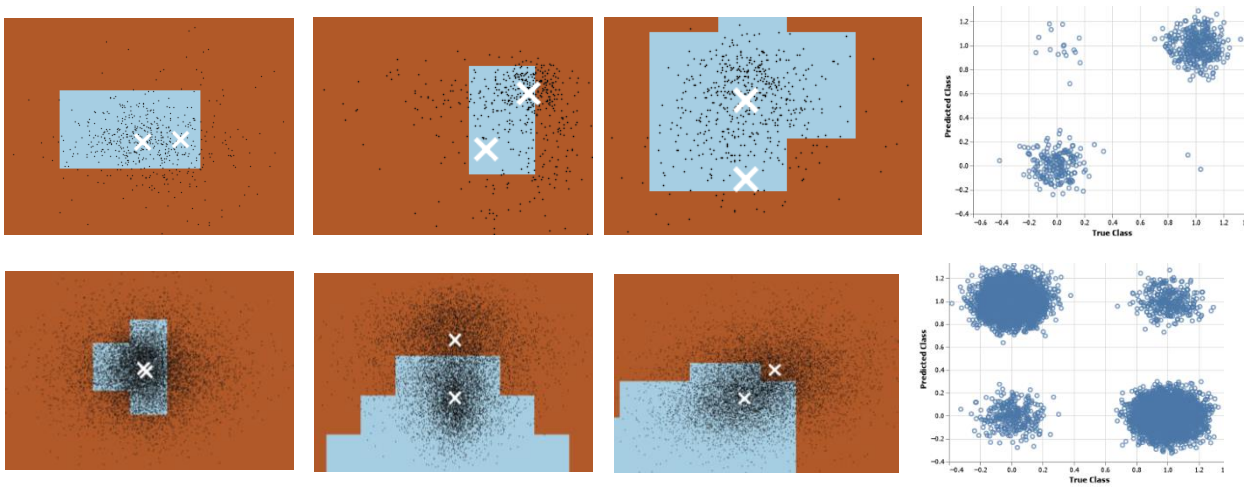




In this portion we will examine all combinations of our datasets, clustering methods and dimension reduction methods, for a total of sixteen tests. We will start by looking at KM and all the possible combinations. From left to right: PCA, ICA, RCA & LDA with the cancer data on top and image dataset on the bottom. It was hard to plot the LDA as it was only 1D, so I created a confusion matrix of sorts with the true class on the bottom and the predicted on the side to show how well LDA was able to separate the clusters and see how well KM was able to classify them correctly. Here are the kurtosis values for each dataset reading left to right, top to bottom: [(3.78,3.7), (16.71,9.05), (4.88,4.12), (-.789)], [(1.547,.589), (1.598 ,-.818), (.59,.518), (-1.08)]

These clusters were spot on, with an average of 93% accuracy for the first dataset and an average of 90% accuracy for the second dataset, the dimensionality reduction techniques were able to split the clusters apart to allow for correct classification. One of the interesting aspects of RCA is its ability to capture a good majority of the information randomly, and we see how it exceled at that. As I ran the RCA algorithm a couple of times on both datasets, I noticed how the clusters would float between what the PCA projects and what the ICA projects. On the bottom, look at PCA then compare it to RCA, the clusters seem further away from a Euclidean point of view and with a silhouette score for RCA at .5 and PCA at .4, it would be safe to assume RCA performs better on the image data than PCA. It did not perform quite as well as ICA though, but that gives us insight into the data. This would lead me to believe the image dataset has features that are better to look at independently than assuming everything is correlated. Consider a picture, there are pixels that are highly correlated, but when you come to the edges of shapes the pixels are independent from each other. These edges are the first things an ICA model will come to detect as they give the most information about what is contained in the image. This is actual a main difference between ICA and PCA. PCA tries to find the average of what ever it is given, the average face, the average size, etc. but ICA can find edges of things because it is not restricted to orthogonality. This tells us that ICA is highly directional and thrives off that type of structure in problems. ICA focuses on independent features while PCA tries to find the principal value that best explains everything. Hmm, it is almost like the answer is in the question!

Cross Examination & EM



Like the last section, we will be examining all the possible combinations with EM. From left to right: PCA, ICA, RCA & LDA with the cancer data on top and the image data on the bottom. Again, due to the bounds of LDA I had to again plot a confusion matrix of sorts, I had to apply a jitter in both the X direction and in the Y direction to create these clusters otherwise it would've been 4 dots on a graph and not meaningful at all. Here are the kurtosis scores left to right, top to bottom: [(3.748,3.708), (16.712,8.939), (11.919,8.969), (-.789)], [(1.547,.589), (1.6, -.818), (.164,.163), (-1.08)]. Please note the reason the kurtosis values are off is because there was no random_state set in the dimension algorithms, otherwise they should've been the same values with the same kurtosis.

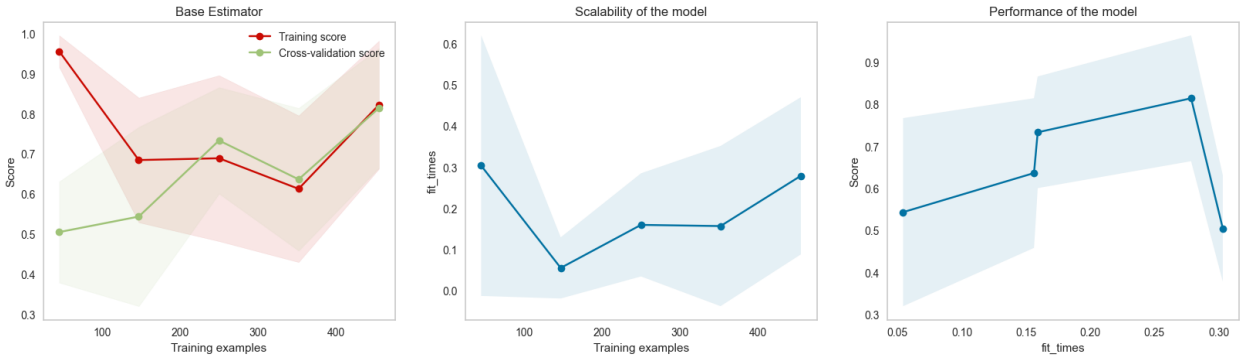
It is interesting to note here that unlike the KM algorithm, EM created a bound all around one area of the graph and then the rest of the graph was left for the other class. The KM algorithm was able to draw a plane, on either side of which fell the classification. It is also interesting to note that on this version, EM had the center of the cluster fall within the same classification on 4 out of the 6 graphs. One might think this is an error, but let's stop and think for a second. The only times this occurred is when EM tried to completely encircle one of the classes (graphs 1,2,3,5). Due to this fact, and the remainder of the other points occurring outside of the bounded area, the center of the outer area has a good chance of falling within the bounded area. Look at graph 5 for an example. It has classified everything inside the circle to be similar and everything outside to be similar. But since the points form an oval like cloud around the inner points the center of the outer points falls almost right on top of the other cluster's center. Another thing to point out is the ICA algorithm, again. Even on a different instance of the ICA algorithm it was able to split the cluster apart which further solidifies the idea that ICA is nothing more than an edge detection algorithm and excels in problems like this. Unlike PCA where it tries to find the 'average' of the image. Again, it is good to point out LDA performed well with an accuracy score of .97 for the first dataset and .94 for the second dataset. LDA would hold its own as a very strong supervised learning algorithm. RCA I would say falls as a middle point between LDA and PCA for these two datasets. It looks at the components

randomly (assuming independence) but is still able to capture information and get close to maximizing the variance (looking for the principal component).

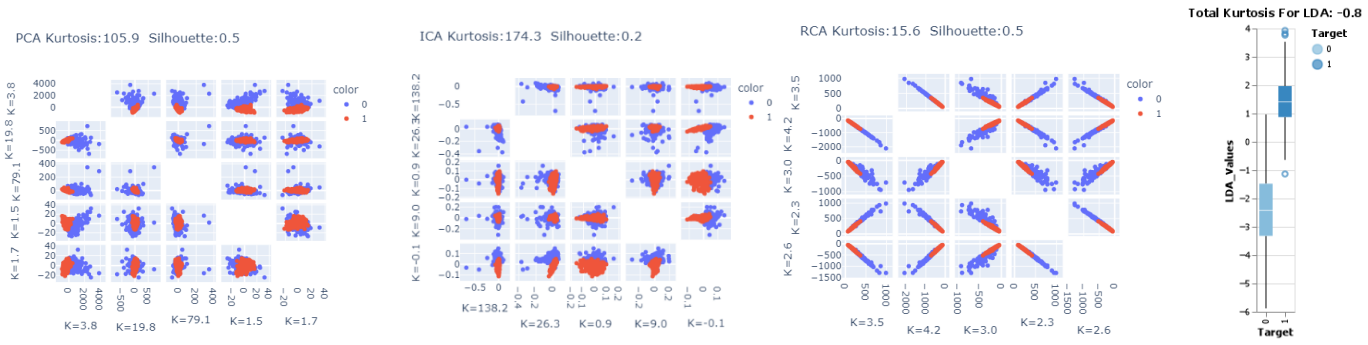
Overall, we have seen the performance of these 4 dimensionality reduction algorithms work in tandem with the clustering algorithms over two distinctly different datasets. I would say ICA with KM performed the best for our image classification dataset and LDA/ICA performed the best for the breast cancer data. That judgement is made not only off the maximized kurtosis values but also how the graphs appeared visually.

Application With ANNs

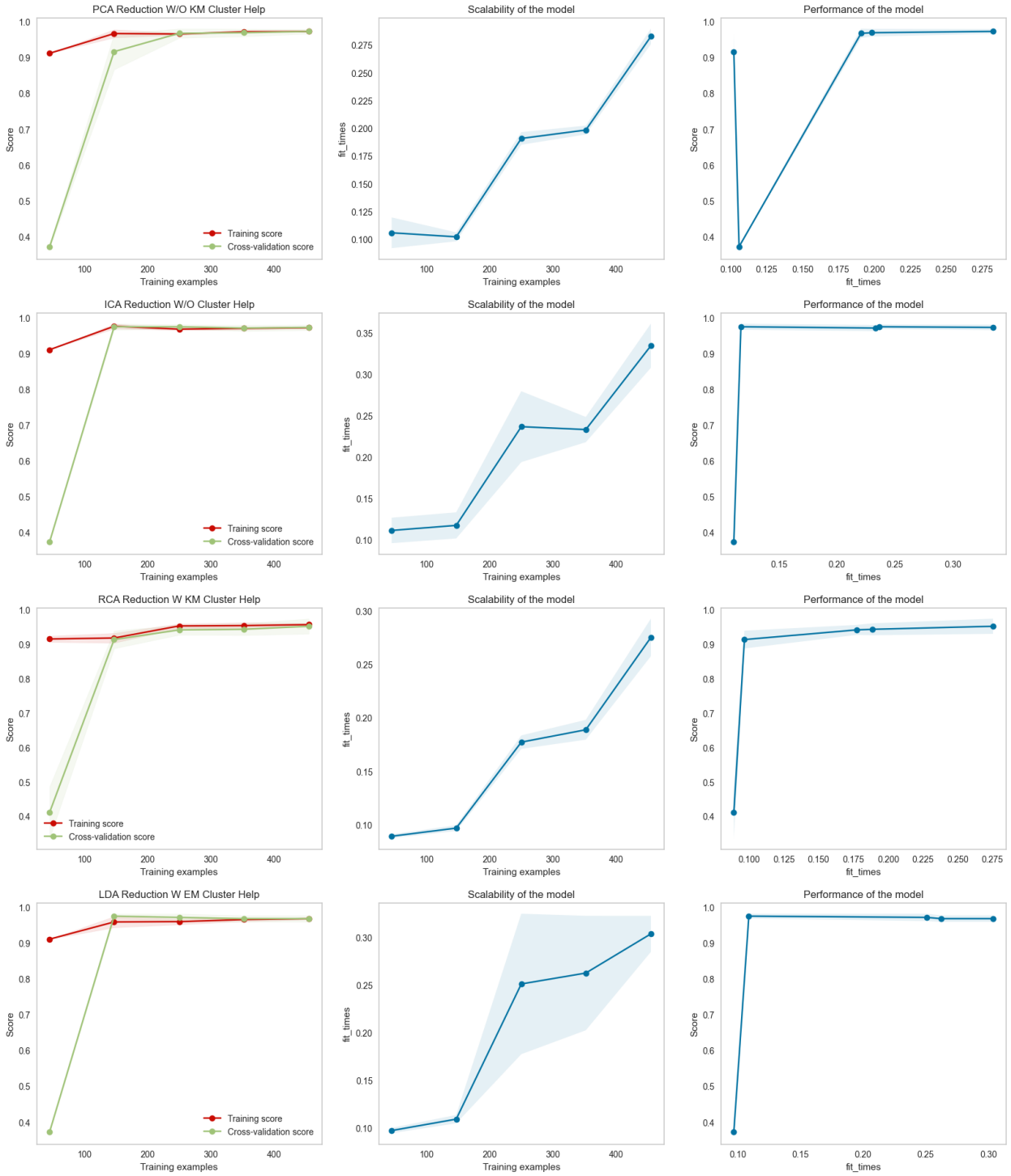
In this portion we will be exploring how dimensionality reduction and clustering algorithms can affect an Artificial Neural Network. First, we will only be looking at dimensionality reduction and how well it performs. Then, we will explore what happens when we use a clustering algorithm to supply its predicted values as a feature in the ANN. We will be performing this analysis on the breast cancer dataset. As a reminder, this dataset contains an unbalanced class of samples with 212 in the negative and 357 in the positive. This means our ANN will lean towards the positive as that is a bias that is present in the data. Below is a reference learning curve for the base estimator using an activation of 'logistic', max_iter = 900, solver = 'adam' and 2 hidden layers. I did a loop through all these possible combinations to make sure these would produce the best model possible.



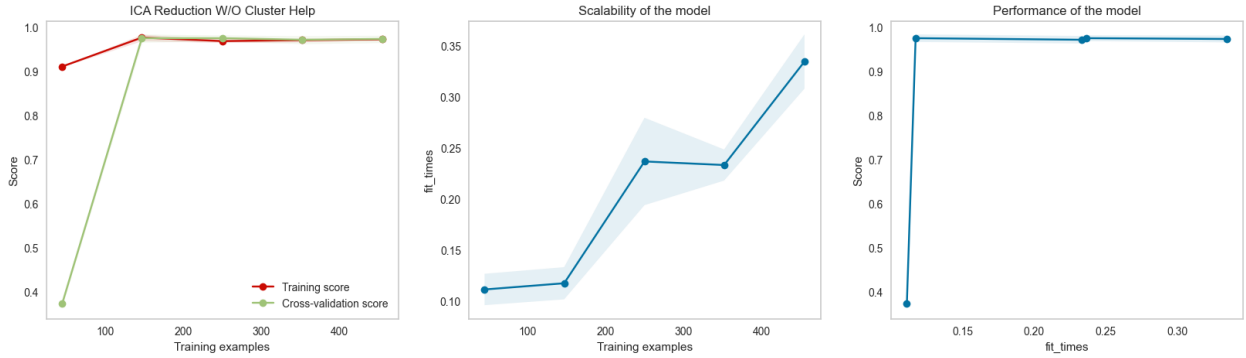
Here we see the model does rather well across the board. The learning curve converges around 85% accuracy, we stay around .3 when training a little lower than 500 samples. Even then though, we can get a good score of 70% with only training on half of our data. Below are the 4-dimension charts to show this decision based on kurtosis for all algorithms.



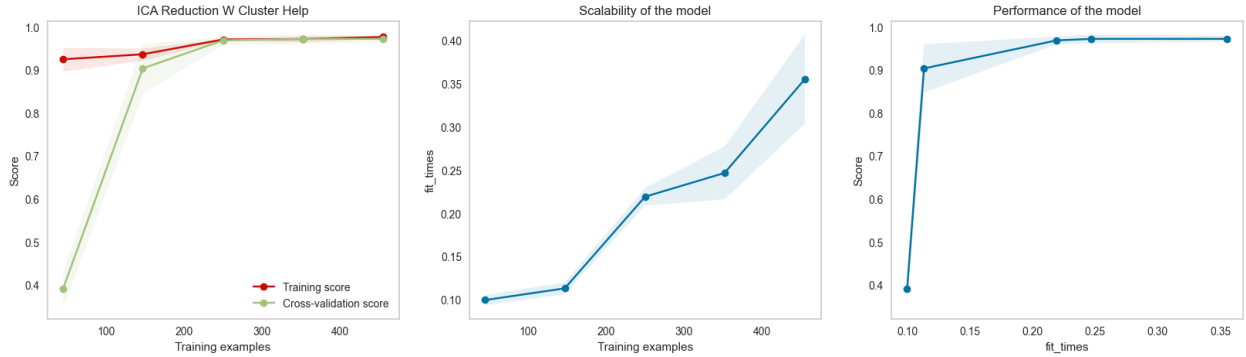
This is going to be a little of a graph spam, but the following are all 4 dimension reduction in combination with the 2 different clustering models and how they perform in the ANN. I only picked the best out of the difference scenarios (Cluster/NoCluster, KM/EM) for a total of 4 graphs. 2 Important things to note, PCA and RCA performed the same across all 4 different scenarios, so I picked on at random because all the graphs looked the same.



Clearly ICA is the best choice with an overall kurtosis of 174 and on the graph, we can see how quickly it achieves a high accuracy score. We will use KM as our clustering algorithm as it seemed to perform better than EM on the cancer data. Now that we have validated our choices, let's see what happens to the ICA algorithm without clusters and with, just to dive in a little more.



Let's start with the graph on the left. Not only did the accuracy jump up to a whopping 97%, but the number of samples needed to get there was almost cut in quarters! Originally, we needed almost all the samples to get up to .85, which is 569, but now we only need about 140-150 samples to reach a higher score. For the graph in the middle the time hovered between .3 and .2 seconds, and here we start from .11 and go up to .35. It made the time possibly exponential, but that doesn't matter because when we look at the graph on the right it shows the fit time compared to the score. As you can tell, in that .11 seconds we are able to jump up to such a high level of accuracy that we don't need to train on any more samples.



Interestingly enough, the model actually does worse. This must mean the KM algorithm is classifying a good portion of the points incorrectly, at least enough to make the performance a little worse. It isn't that bad though; we go from an accuracy of 97% at 150 data points to 90 at 150 data points and we approach the 97% with an extra 100 datapoints for a total of 250 data points. The scalability of the model is more linear than the other graph, so if we had a problem with performance, we could look at this as an option. It is interesting how too much information can hurt a model's performance.

Works Cited

Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

Kurtosis. Kurtosis - an overview | ScienceDirect Topics. (n.d.). Retrieved November 5, 2022, from https://www.sciencedirect.com/topics/social-sciences/kurtosis