Analysis of Multiple Clustering & Dimensionality Reduction Algorithms

For this report I will be analyzing the effectiveness of multiple clustering and dimensionality reduction algorithms on the breast cancer and user knowledge modeling datasets provided by the UCI Machine Learning Repository.

1.0 Breast Cancer Dataset

I chose this data because breast cancer is a common occurrence, and a misdiagnosis could be fatal. If a false positive for a malignant tumor occurs, then it could lead to a person getting unnecessary chemotherapy, while if it's a false positive for a benign tumor, then they could live with breast cancer until it becomes a far bigger problem. It would be interesting to see how effective a machine learning algorithm could do on a set like this.

The breast cancer dataset contains 699 instances. The researchers noted 10 features, and labeled whether the tumor detected was benign (not harmful) or malignant (harmful).

Before starting with the data analysis, I cleaned the data to only focus on the features that would give a good prediction of the type of cancer. To do this, I removed the samples with missing information (containing '?' values). There were very few instances of this, only 16. This reduced our data set size to 683 instances. After this, I removed the "Sample Code Number" feature, as it is simply an id that has no relation to the actual cancer prediction. This reduces the number of features we are looking at to 9.

1.1 User Knowledge Modeling Dataset

This dataset contains features that correlate to patterns of study for a participant. I found this interesting because it can showcase the different methodologies of studying and how effective they might be on the overall knowledge of the participant. This is applicable to school life as it would help improve education and learning.

The dataset contains 403 instances, and 5 features, relating to study time, repetition of material, study time with related objects, and two exam performances. The target variable is a value relating to the participant's knowledge level which is measured as a categorical variable between "very low" and "very high".

For both datasets, I took out 20% of the data for testing, while using the rest for training and cross validation.

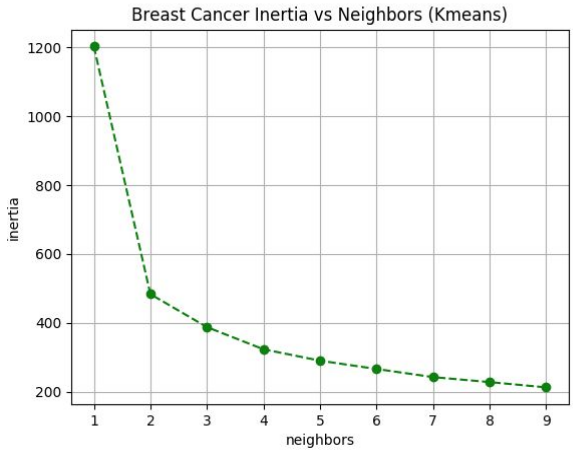
2.0 Experiment 1 (Running K-means and GMM)

In this experiment I ran the clustering algorithms (K-means and Gaussian Mixture Models with Expectation Maximization) on the two datasets and noted interesting features about them. For all the clustering algorithms, I ran them from 1 neighbor/gaussian up to the number of classes for the dataset. I did this so I could note how the clustering changes with the number of neighbors, or possibly find a clustering that works the best or is interesting. I did not try more neighbors or gaussians because the model complexity should ideally be as simple as possible, and I can reasonably expect that the gathered features themselves are the most complex.

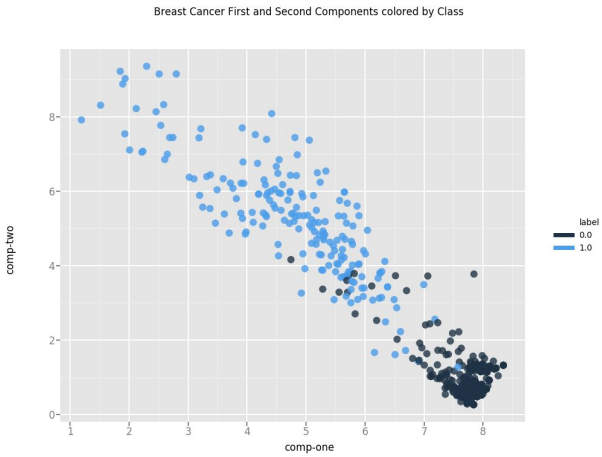
2.0.1 Breast Cancer Data

K-means:

For this data, I noted the inertia (sum total distance away from the cluster centers) for every neighbor I tried out. Along with this I plotted the relationship between the first two variables to see if they were separable. The cross section would let me visualize a segment of clusters, and if they seem separable, they should ideally be the centers of individual clusters.



Img 2.01: Breast Cancer Inertia vs Neighbors (K-Means)



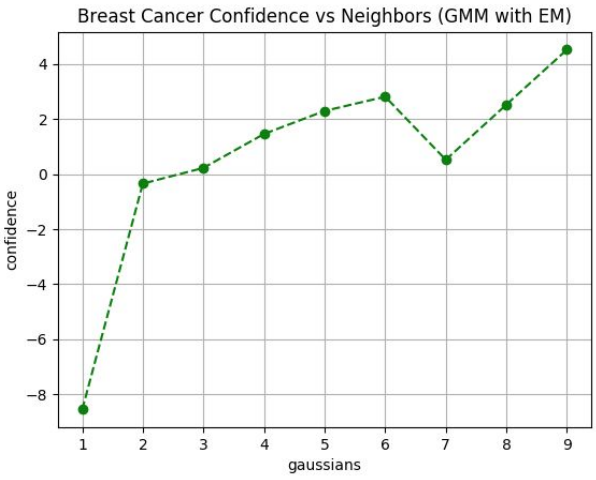
Img 2.02: Breast Cancer Component Cross Section (K-Means)

We can see that more neighbors generally tends to lower inertia, which makes sense. This is because as we get more clusters, the clusters will get closer to individual points in the feature space. The space between them will reduce, and result in a strictly decreasing inertia.

On the right we see a cross section of the first and second components for the 8 neighbor case. This cross section shows how well our data set is separable. In the right bottom side there is a clear cluster of points, which tend to be high comp-one values, while comp-two takes the lower values. What does that mean? It means that in this case, the K-means algorithm has likely been able to cluster the data points well enough to get a good separation between variables. The newly formed clusters may have better information than the direct classes themselves in terms of finding latent (hidden) correlations. As we can see there is a sliver of dark blue on the horizontal at comp-two equals 4. This is something the algorithm wasn't able to pick up on, but another cross section might have.

GMM:

Img 2.03: Breast Cancer points vs gaussians (GMM)



Img 2.04: Breast Cancer Component Cross Section (GMM)

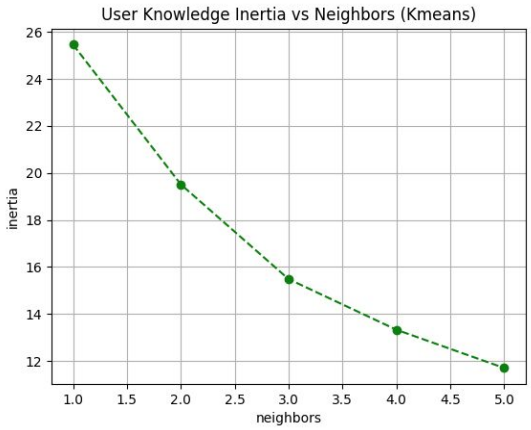
With Gaussian Mixture Models, we try to find gaussians that explain our data well. I ran through all the gaussians from 1 to 9, but picked out the 7 gaussians one in particular because it appears that 3 of the gaussians

were able to explain over 400 instances. What does that mean? If GMM was able to find 3 highly correlated points, then there is likely a strong dependence on these variables to the label class or output, whether that's on one feature or multiple features. What I'd expect to see is that if we were to do dimensionality reduction, these features would be the first to go (in this case gaussian 0). Along with this, these 3 gaussians likely have the most influence on the final output class.

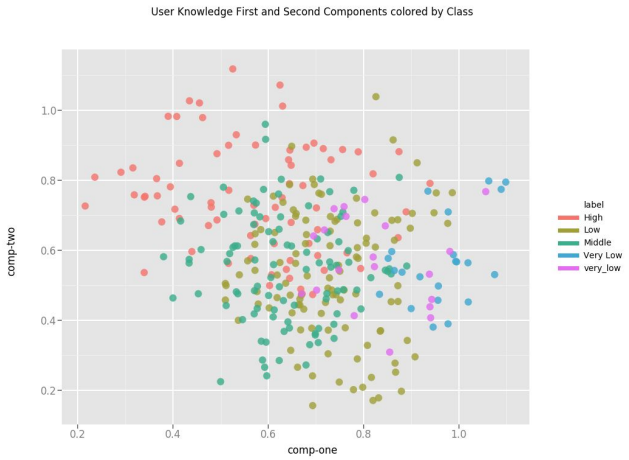
2.0.1 User Knowledge Data

K-means:

Similar to the breast cancer data set, I'm trying to find some interesting features in the data by going through all the numbers of neighbors from 1 to the number of classes.



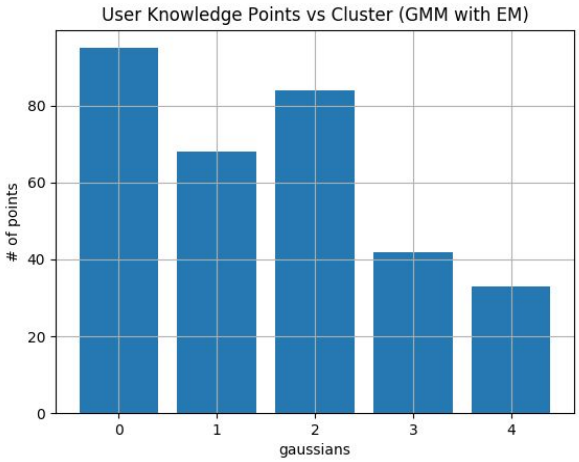
Img 2.05: User Model Knowledge Inertia vs Neighbors (K-means)



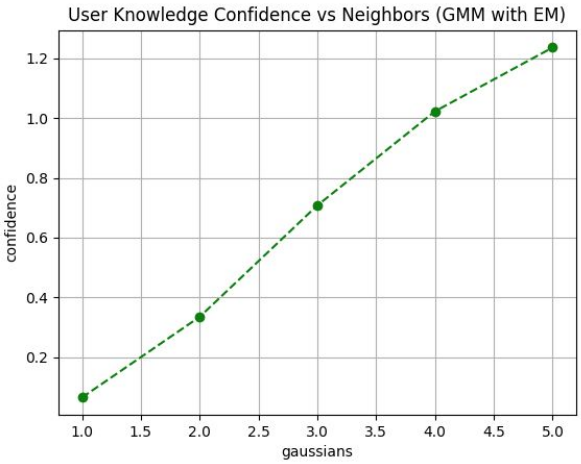
Img 2.06: User Model Knowledge Component Cross Section (K-means)

Here we see the graph of inertia, which is again strictly decreasing, and the cross section of the first features. From this cross section we can see that the data is quite jumbled. It doesn't appear to have any clear clusters, which would lead me to think that this would be a particularly bad set of features to use because it'd likely fair no better than random guessing.

GMM:



Img 2.07: User Knowledge Points vs Gussians (GMM)



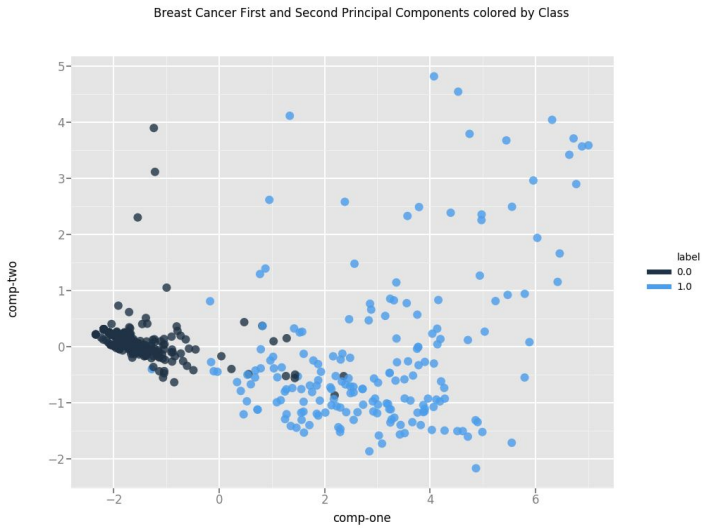
Img 2.08: User Knowledge Confidence vs Gaussian (GMM)

For GMM, we can confirm what we saw in K-means, that is that the features are not easily separable. Particularly, if we look to the graph on the left, which depicts the number of points in each gaussian, we can see that there is a relatively even distribution, with no real one dominating the feature space. This leads me to believe that this dataset likely has some latent variables that weren't taken into account by the data provided.

3.0 Experiment 2 (Running PCA):

For this experiment I'll be applying Principal Component analysis to the two datasets and recording the results. I ran the PCA algorithm until it was able to explain 95% of the variance in the dataset, which means that we should expect to see fewer features from our principal components than in the data sets. After running PCA, I took a cross section of the data with the 2 most popular features so as to compare them to K-means and GMM, to see if PCA was able to reduce the dimensionality as I had expected.

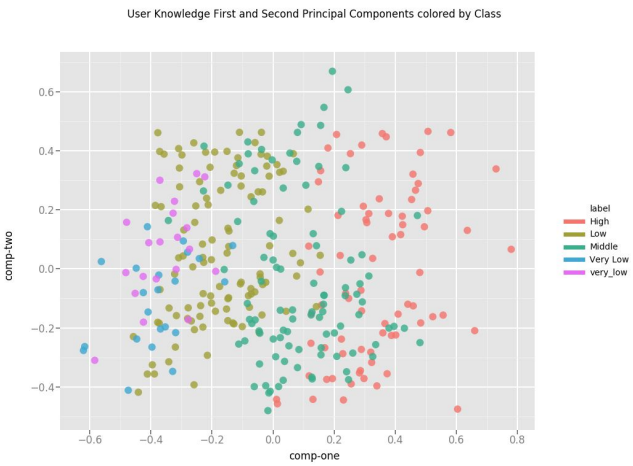
3.0.1 Breast Cancer Data:



Img 3.01: Breast Cancer Principal Components Cross Section (PCA)

After running PCA, the two most explained components are plotted above. As we can see, these components seem to be well separable and have an explained covariance of 0.6520434 and 0.08269258. This means that comp-one explains about 65% of the features, while comp-two explains about 8%. Recall the cross section from section 2.0.1, which, when compared to this cross section, seems to better linearly separate the data. From this cross section, we can almost draw a vertical line at comp-one equals 0, to separate most of the labels. It looks like PCA was able to more tightly compact the clusters, which means that it might improve our features.

3.0.2 User Knowledge Data:



Img 3.02: User Knowledge Principal Components Cross Section (PCA)

From this graph we can see that the points are similarly difficult to separate. The total number of principal components PCA gave me was 5, which is also leads me to believe that there was difficulty separating them since that's the same number of classes we have. Our principal components seem to not have a clear distinction between all the points, and this is explained by the covariance, which is 0.30013953, 0.23411406, 0.19765882, 0.1488235 , 0.11926409. Essentially very similar covariances and not a lot of distinction between points.

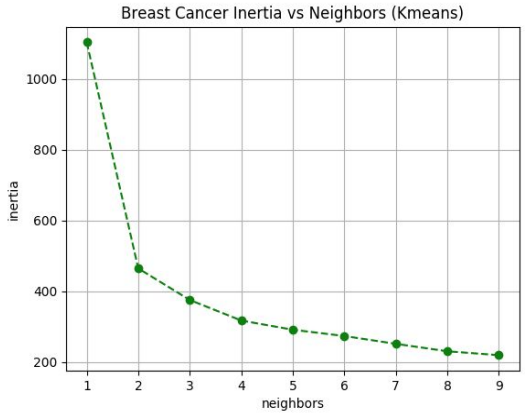
This makes a good deal of sense because 5 features is probably not good enough to explain accurately how humans learn. There is a lot of variance between individuals themselves, and likely a lot more variance in how each method was applied. This dataset is likely not going to show nuances in the data, even though we attempted to preserve the features, or find better features.

4.0 Experiment 3 (Clustering after PCA)

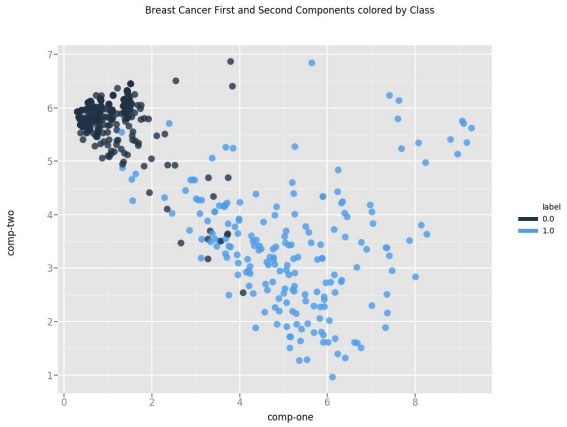
To see if it's possible to get better cross sections/segmentation, I'll be applying clustering algorithms to the PCA data. The expectation is that we'll see fewer features (reduced dimensionality), and that might lead to better clusters because the data is transformed. I ran the same metrics, so we'll be able to do a direct comparison to the features in experiment 1 (Section 2.0).

When comparing the post PCA clustering to normal clustering, we should expect to see reduced dimensionality, an axis flip (because PCA calculates eigenvectors and orthogonal projections), and possibly tighter clusters.

4.0.1 Breast Cancer Data:

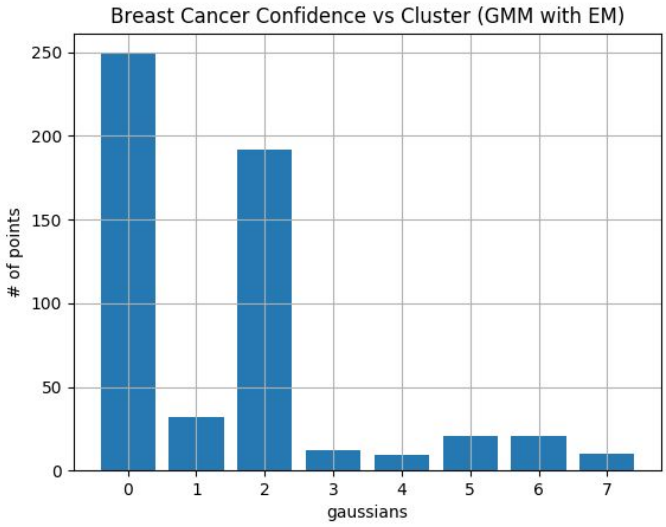


Img 4.01: Breast Cancer Inertia vs Neighbors (K-Means with PCA)

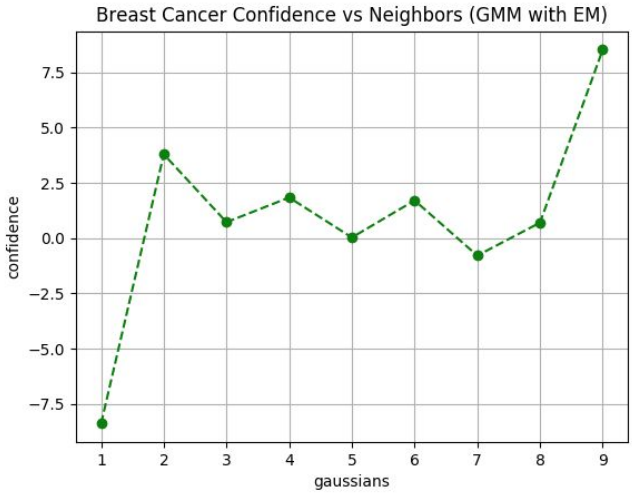


Img 4.02: Breast Cancer Component Cross Section (K-Means with PCA)

Looking at the inertia, the values are nearly identical with experiment 1. When looking at the component cross sections, we can see there is the axis flip. Along with that we see a much tighter cluster for the 0 and 1 labels. However, likely because of PCA, we can see that there is more merging between the outliers of the datasets. This makes sense because when getting the principle components, the eigenvectors aren't 100% accurate. They will end up getting projection that is good enough, but will end up shifting nearby points away. This may affect the final outcome to a smaller degree (something to keep note about when running against neural networks in later experiments).



Img 4.03: Breast Cancer Points vs Gussians (GMM with PCA)

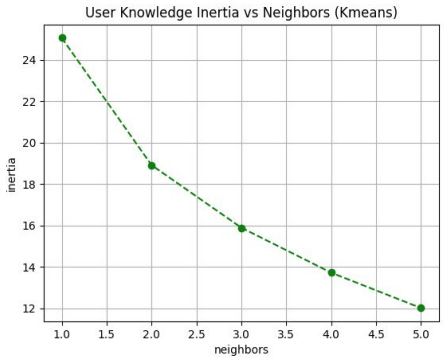


Img 4.04: Breast Cancer Confidence vs Gaussian (GMM with PCA)

For GMM, we can expect that PCA would try to preserve the best features, which means that it would accentuate existing clusters. This is exactly what we see in the left graph of gaussians vs. the number of points they occupy. From this we can see that 2 gaussians--0 and 2--dominate the feature space by representing over half the data points. These two gaussians would likely make for good features and do a lot of the classification in a supervised learning setting.

4.0.2 User Knowledge Data:

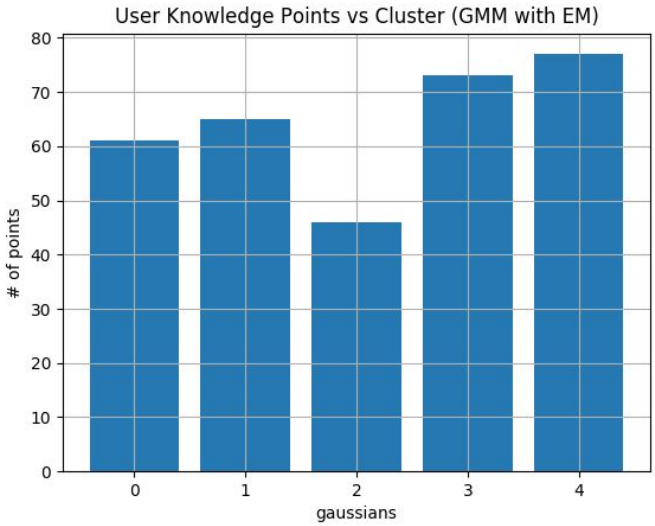
For the user knowledge data, I wouldn't expect there to be a huge difference, because we established that the data was already difficult to separate in the PCA experiment.



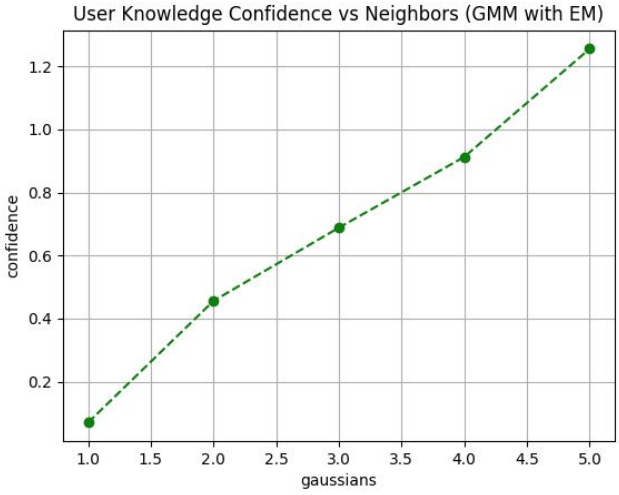
Img 4.05: User Knowledge Inertia vs Neighbors (K-Means with PCA)

Img 4.06: User Knowledge Component Cross Section (K-Means with PCA)

As expected, the inertia strictly decreases, and the component cross section seems to remain inseparable. PCA flipped the axis, but overall the variance appears to have remained roughly the same. K-means with PCA wouldn't suddenly improve the classification if the data itself is bad or inseparable.



Img 4.07: User Knowledge Points vs Gussians (GMM with PCA)



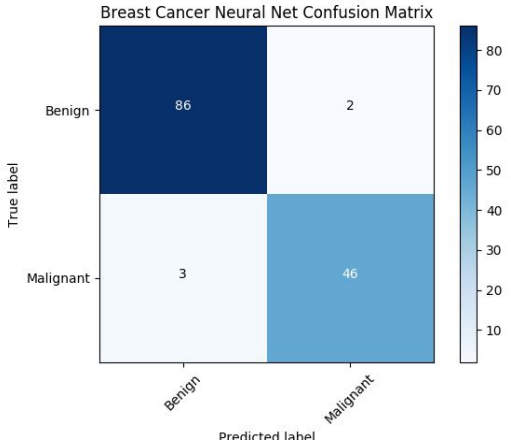
Img 4.08: User Knowledge Confidence vs Gaussian (GMM with PCA)

Similarly as in experiment 2, there appears to be no truly distinguishing gaussian amongst the dataset. In fact, the variance has decreased with GMM, which might make performance with PCA and GMM even worse when trained with a supervised learning algorithm. This is because PCA tried to find the eigenvectors that best represented the data, and ended up overgeneralizing in a sense. The gaussians would try to model the PCA distribution, and end up with a lower variance than if the data just had run on the dataset raw.

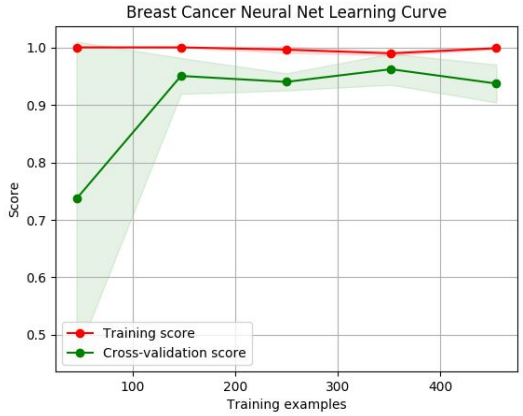
5.0 Experiment 4

In this experiment I'll be using the breast cancer dataset I've applied PCA to, and try training a neural network on it. As we've seen in the previous few experiments, I'd expect PCA to perform about as well, or slightly better than the baseline, because it managed to find a good separation and tighten the clusters.

5.0.1 Breast Cancer Data (Control)



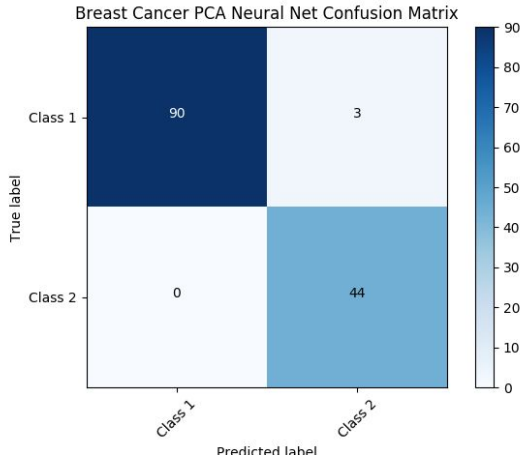
Img 5.01: Neural Net Confusion Matrix



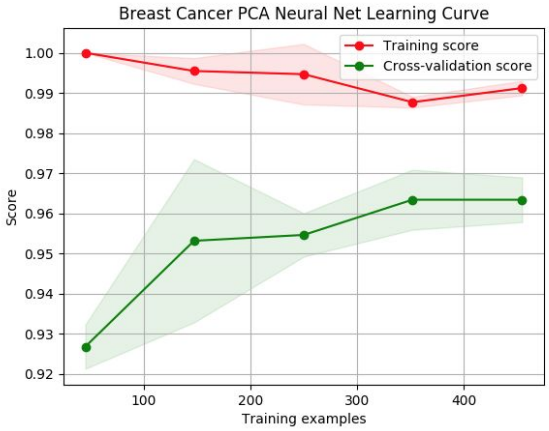
Img 5.02: Neural Net Learning Curve

In the control, we can see that the neural net only had 5 incorrect markings, which resulted in a 96.35% accuracy.

5.0.2 Breast Cancer Data (PCA)



Img 5.03: Neural Net Confusion Matrix (PCA)



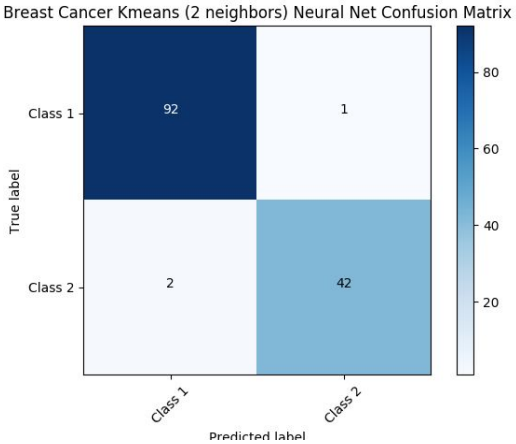
Img 5.04: Neural Net Learning Curve (PCA)

With PCA applied, the neural network was able to get a slightly higher accuracy than before, only getting 3 incorrect guesses. This is a 97.81% accuracy. This is expected, because as we saw in experiment 2, the clusters had tightened, and separation was better. This means that the neural net was able to select the better principle components that describe the data better and do slightly better than the regular classes. This also suggests that there is some relationship between variables that could influence the resulting label.

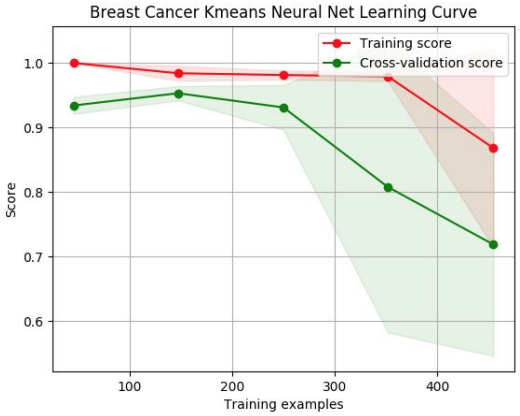
6.0 Experiment 5

For this experiment I'll be running the clustering algorithms done in previous experiments and feeding the transformed features into a neural network to see how well they perform. I ran these algorithms again from 1 neighbor/gaussian to the number of classes.

6.0.1 Breast Cancer Data (K-means)



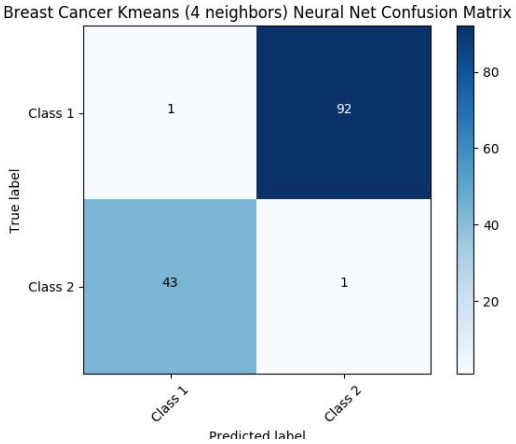
Img 6.01: Neural Net Confusion Matrix (K-means)



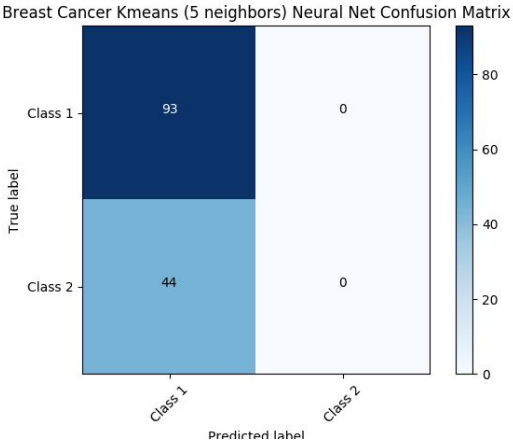
Img 6.02: Neural Net Learning Curve (K-means)

K-means had a very surprising result as its best score was able to match that of PCA and beat the baseline, scoring an accuracy of 97.81%. This means it was able to find a clustering that was better than the given data, as speculated in experiment 4.

However, upon further inspection it became evident that this might have been a lucky cluster.



Img 6.03: Neural Net Confusion Matrix Flipped (K-Means)



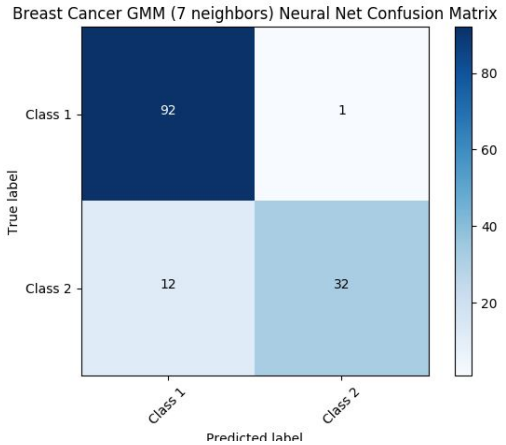
Img 6.04: Neural Net Confusion Matrix Bad (K-Means)

On the left we can see that K-means was really good at putting the classes in the wrong section, but this would be an easy fix since it's binary, we'd just flip the label. This clustering could have gotten a 98.54% accuracy.

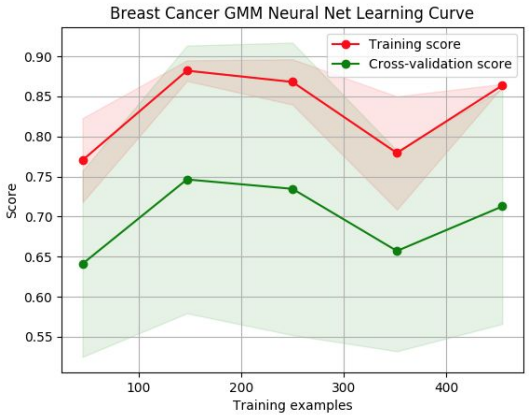
On the right we can see a very poor clustering where every point was labeled as "Benign". This was likely a bad clustering which K-means is susceptible to if, for example, a centroid happened to be between two similarly sized distributions, and hence make a poorly segmented new feature space.

6.0.2 Breast Cancer Data (GMM)

For GMM I would expect that the result would be a slightly noisier version of the baseline neural network because of dependence.

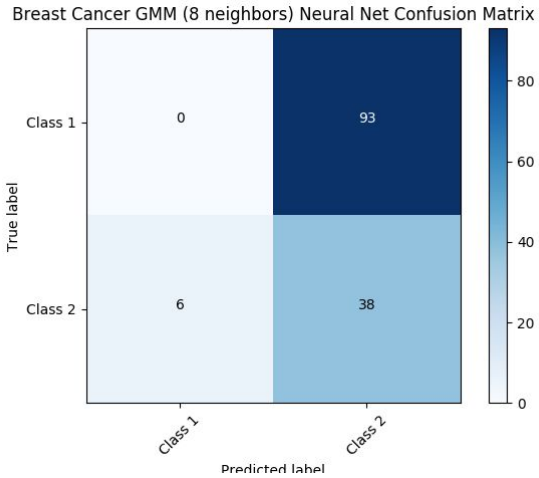


Img 6.05: Neural Net Confusion Matrix (GMM)



Img 6.06: Neural Net Learning Curve (GMM)

The GMM feature neural network ended up doing worse than the baseline, this might be because the gaussians selected were in between good clusters, or the features they clustered on took in too many outliers of the opposite class.



Img 6.07: Neural Net Confusion Matrix (GMM)

This confusion matrix shows that even with an increasing number of neighbors, the prediction can get significantly worse as well.

The upshot to using GMM is that it was significantly faster to run on my computer than the other algorithms.

7.0 Conclusion:

Overall, doing K-means on the datasets seemed to be effective but had a high variance (experiment 5). It was able to find a good cluster in the data, and that cluster became a good feature that helped distinguish the labels.

GMM was good at identifying what features tended to cluster together (experiment 3), but that did not necessarily result in good neural network results (experiment 5). The advantage to GMM however is speed as it is much faster to put the feature into a gaussian than other algorithms which require multiple calculations.

PCA was good when it came to easily segmentable and datasets, or ones that had likely latent correlations (experiment 4), and sometimes made no difference (experiments 2 and 3) in the classification. PCA is a solid choice overall, but could end up adding a little bit of noise to the features because its decomposition isn't always 100% accurate, though it's able to get the eigenvectors that most represent the data.

Citations:

UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/index.php

Scikit Learn: http://scikit-learn.org/