Peter Lee
Professor Byron Boots
Username: plee99
CS 4641: Project#3 Unsupervised learning.
11/20/2018

Introduction:

For this Project Scikit Learn (Sklearn), a python library, was used to solve most of my machine learning problems. In this project, two different clustering algorithms were implemented on data: K-means Clustering (**K-Means**) and Gaussian Mixture Model (**GMM**). K-means Clustering assigns each point to a cluster that's closest to it. K random points are initialized and the Euclidean distance from the data nearest to each point, and "claims" it. The each centroid is calculated and the centroid moves closer to mean of all the data that it "claimed." Then the previous steps are repeated until the clusters converge and the cluster locations stop changing. Sklearn's Kmeans Inertia attribute was used to evaluate each model. Inertia is the sum of the squared distance from each data point to the nearest cluster.

Gaussian Mixture Model via expectation maximization (GMM): This algorithm assigns probabilities to each data point with the likeliness to belong to a particular cluster. GMM's Bayesian Information Criterion (BIC) in Sklearn was used to evaluate each model. BIC is related to the log likelihood function as well as including a penalty term for the complexity of model to prevent overfitting. The lower the BIC, the better.

Datasets:

The two datasets that I chose for this project were the Breast Cancer Wisconsin (Diagnostic) Data Set and the Sloan Digital Sky survey data set. Both of these problems are classification problems with multiple variables. The datasets were both sufficiently large enough (both over 1k samples) to be split into a training and testing set. The testing set was never used to train the model and was used as a metric to estimate the overall effectiveness of the learning algorithms. Splitting the data ensured that information contained in the training set was enough for the model to generalize the information to correctly classify and predict the labels of the test set, which the machine learning model has never seen. The training and testing set were consistent between all of the various machine learning models to minimize variations in the models. The datasets were the following:

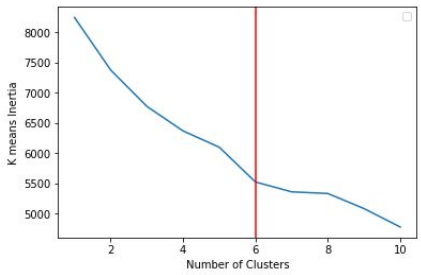
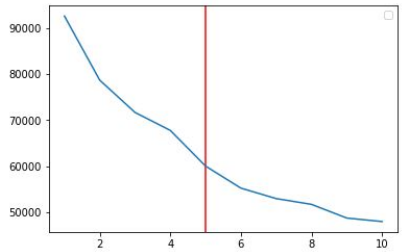
Breast Cancer Wisconsin (Diagnostic) Data Set - Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass.They describe characteristics of the cell nuclei present in the image. [1] The 29 features were associated with a label that indicates whether the breast contained a benign or malignant tumor. The features describes various statistical measures associated with the image of the breast mass such as the symmetry, concavity, and area of the region. The data was preprocessed from images and the dataset was a compilation of various statistics from the images. The data was scaled before implementing it in the code.

Sloan Digital Sky survey data set - The dataset consists of 10,000 observations of space taken by the Sloan Digital Sky Survey. Each observation is described by 15 feature columns and 1 target column which identifies the observation to be either a star, a galaxy or a quasar. [2] The dataset has multiple variables as well as 3 possible labels. The data was already preprocessed from images prior to use.

Clustering

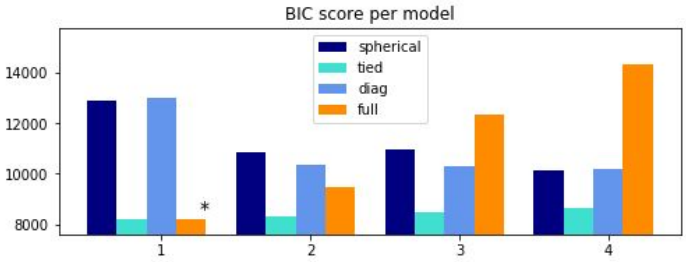
K-means Breast data:

K-means clustering was applied to the Breast Cancer dataset. Inertia was used as a metric to determine the performance of each clustering model. The optimal number of clusters was determined by taking the “Hinge” or “knee” where the algorithm begins to have diminishing returns.

K means Breast data: The “Hinge” is very slight and the inertia begins to gradually decrease. The optimal number of clusters was 6.	
K Means Space data: The “Hinge” is much more dramatic than the Breast Cancer data, It appears the optimal number of clusters is 5.	

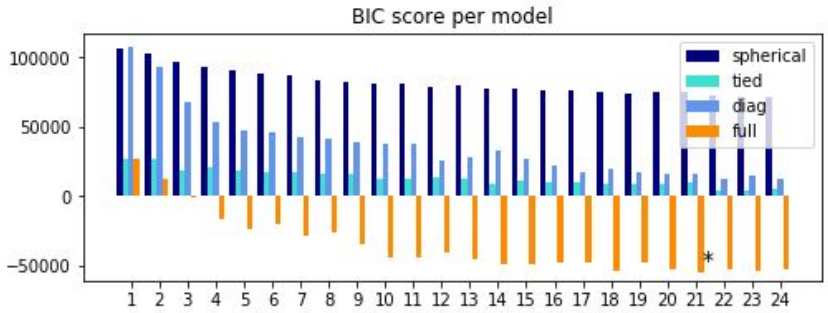
Breast Data GMM:

The figure below shows that the simpler “Full” GMM model was the best model for this dataset. The “full” Model is where each component has its own general covariance matrix relating to each cluster. The data was trained on a training set and was evaluated on a separate test set to ensure the legitimacy of the performance. Each different kind of GMM was iterated through to find the optimal model in terms of BIC.



Space Data GMM:

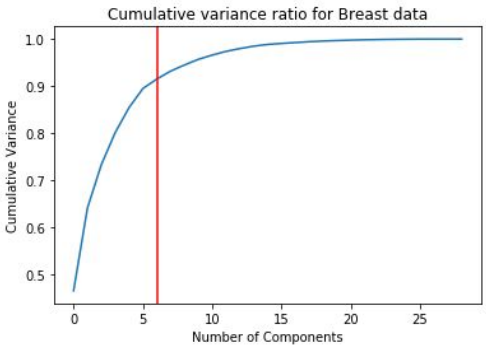
The figure below shows that a more complex “Full” GMM model of over 21 components was the best model for this dataset. The “full” Model is where each component has its own general covariance matrix relating to each cluster. The data was trained on a training set and was evaluated on a separate test set to ensure the legitimacy of the performance. This is most likely because of the difficulty in trying to completely separate the data as demonstrated later when PCA is applied to the dataset and plotted in 3 dimensions.



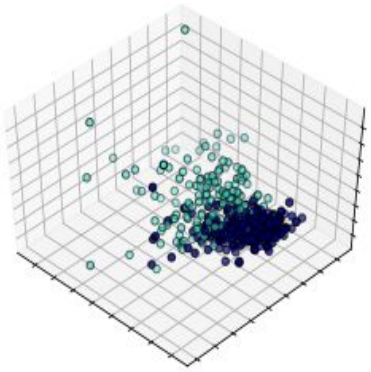
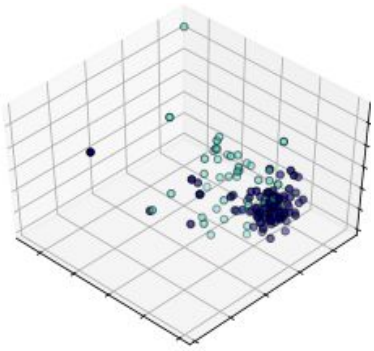
PCA: Principal Component Analysis

Breast Cancer data PCA:

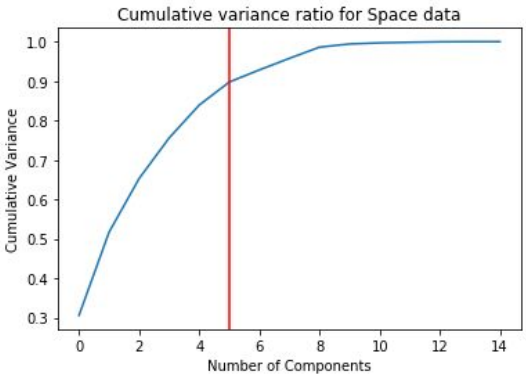
PCA was applied on the Breast Cancer data, the cumulative variance over the amount of components was plotted to demonstrate how increasing the number of components increases the overall “information gain” indicated by the variance since the point of PCA is to maximize the variance. The same algorithm to find the “knee” or “hinge” from k-means was reapplied to find the point of diminishing returns and plotted that as the optimal number of components.
The “Hinge” was around 6 Components. We will use this later and apply these to a model where we first use PCA to reduce the dimensionality of a dataset and then feed it into a neural net.



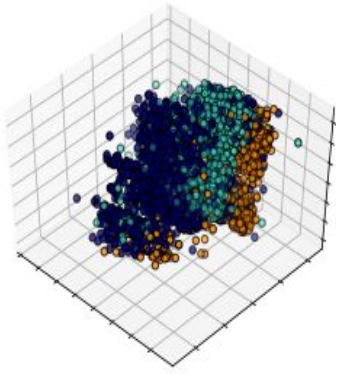
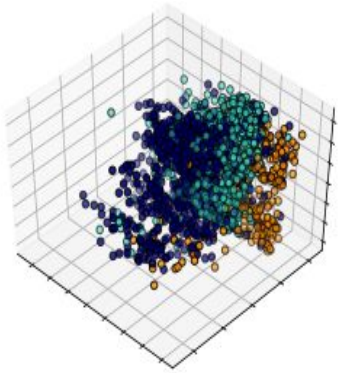
PCA to produce a 3D visualization of the Breast Cancer dataset

Using PCA the number of dimensions of the breast cancer dataset was converted from 29 features to 3 features. All of the data with respect to these 3 features was plotted and colored them with respect to their label whether they were malignant or benign tumors. The close proximity of each data point can be clearly seen indicating that clustering was effective, however there is clearly "mixing" of the clusters as certain data points of the opposite label mesh together in both the training and test data. This demonstrates the inherent inseparability within this feature space. The data was reconstructed very well as we can see. PCA was first fit to the training set then generalized to the test set.	Training Data 	Test Data 

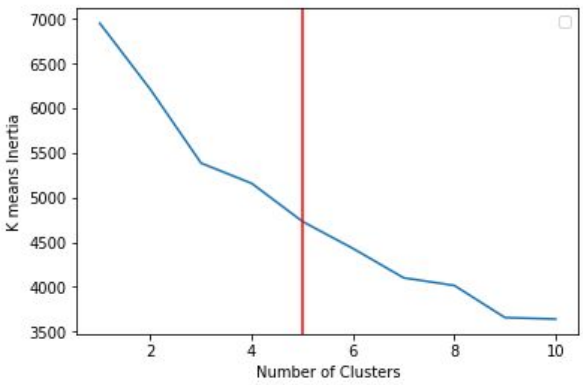
Space Data PCA:

PCA was applied on the Space data, the cumulative variance over the amount of components was plotted to demonstrate how increasing the number of components increases the overall "information gain" indicated by the variance since the point of PCA is to maximize the variance. The same algorithm to find the "knee" or "hinge" from k-means was reapplied to find the point of diminishing returns and plotted that as the optimal number of components.	

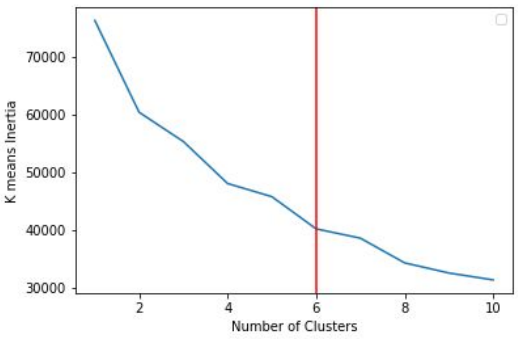
PCA to produce a 3D visualization of the Space dataset

After PCA was implemented on the Space data set, the dimensions were converted from 15 features to 3 features, each data point was plotted in the figure. Each data point was colored with respect to the 3 possible labels. Separation between the different labels is clearly visible. Data points that are mixed together with other data points of opposite labels is also clearly seen. These will most likely be points that are indifferentiable from their neighbors and show that this dataset is not completely separable (at least in this model). The data was reconstructed very well as we can see. PCA was first fit to the training set then generalized to the test set.	Training Set 	Test Set 

Applying PCA then K Means Breast data:

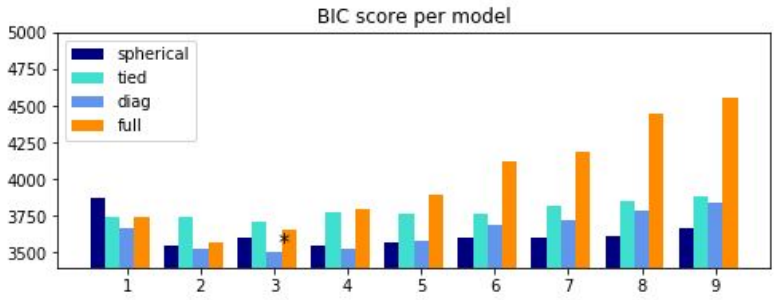
After using PCA to reduce the data down to 6 components. I ran K-means clustering on the resulting 6 features and plotted the K-means Inertia as a function of the number of clusters. The Inertia was lower then just doing K-means clustering on the dataset(~5k vs. 5.5k) indicating that PCA was able to improve the performance of K-means clustering within this instance, Most likely by “tightening” up the clusters.	

Applying PCA then Applying K-means on the Breast Cancer Dataset.

After using PCA to reduce the data down to 5 components, K-means was applied to the data and the “hinge” was found. Overall the Inertia at the hinge was lower than the K-means algorithms without PCA (~40k vs. ~60k). This is a sign that PCA was able to improve the performance of K-means clustering in terms of Inertia, meaning the clustering was nearer to each centroid.	

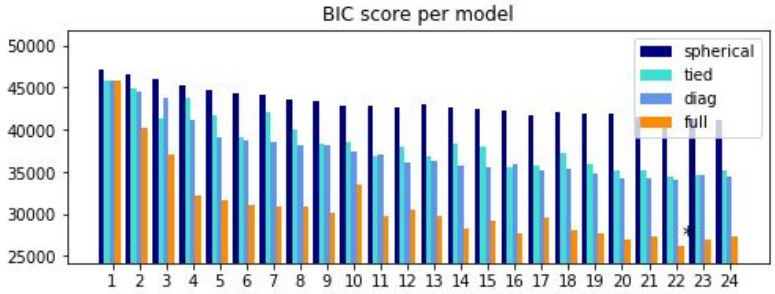
Applying PCA, then Applying GMM on the Breast Cancer dataset:

PCA was applied to reduce the dataset to 5 components, the “hinge” or point of diminishing returns. The BIC scores was lower in this graph than when GMM was applied on the Breast Cancer dataset without PCA. This is probably because PCA was able to ‘tighten’ the clusters and allowed for better training.



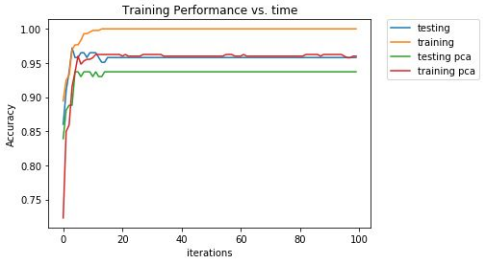
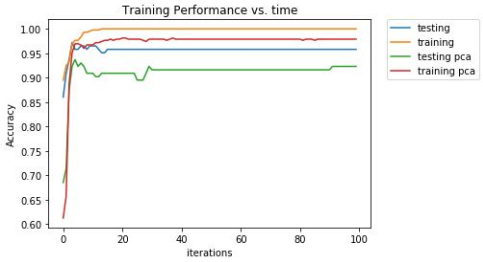
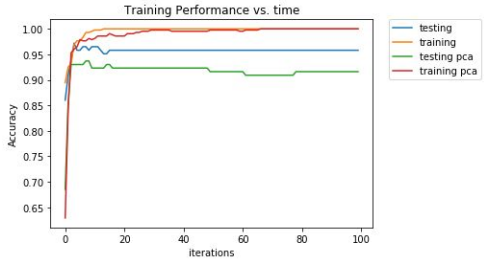
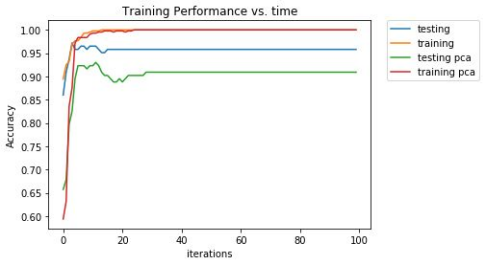
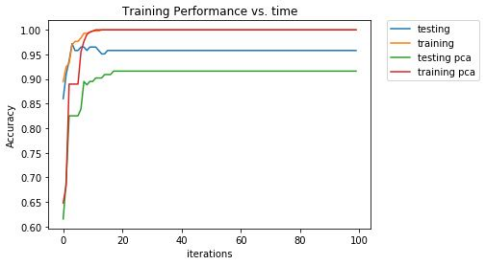
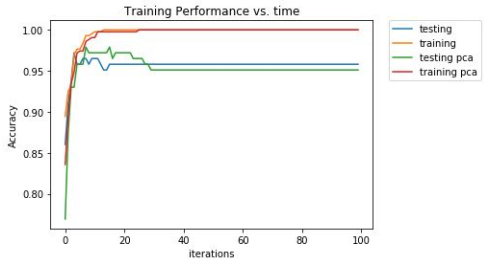
Applying PCA, then Applying GMM on the Space dataset:

After PCA, the GMM of the space data set does worse than without PCA. PCA reduces the number of dimensions of the data and therefore there is some relative information loss as a result of decreasing and ‘squeezing’ the dimensionality, although doing this greatly simplifies the model. As we’ve seen the Space dataset is difficult to perfectly separate and each variable was probably very important to tease out the true model. As a result, the BIC scores were much higher than in the graph without PCA on it.



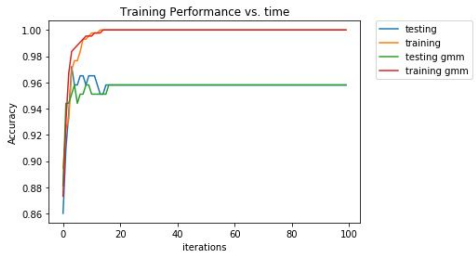
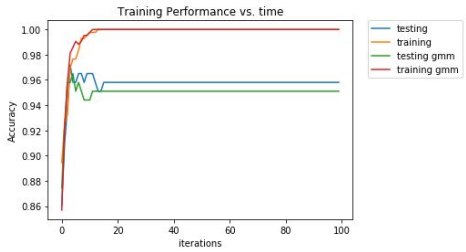
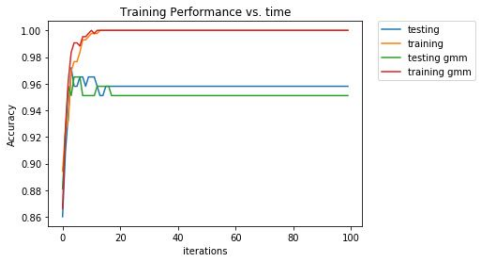
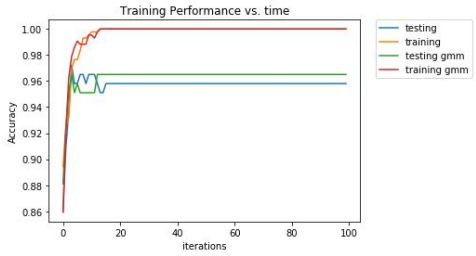
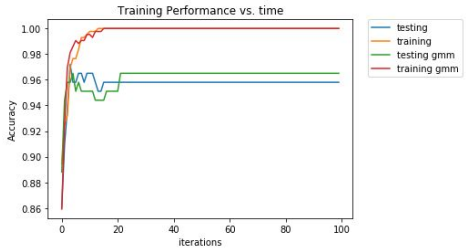
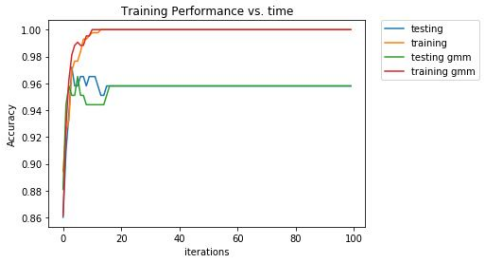
Applying PCA, then feeding into a Neural Net on the Breast cancer dataset:

Overall applying PCA to reduce the dimensionality reduced the training time in relation to how drastic the dimensionality reduction was however It hurt performance. This was most likely since it reduced the importance of certain variables that helped the performance in slight but meaningful way. When the PCA results were concatenated into the training data, the performance of the neural net improved early in the training. If an early stop was used on the training, the neural net with PCA added to the data would have outperformed the neural net with just the training set data.

reduced to 2 attributes 	reduced to 3 attributes 	reduced to 5 attributes 
reduced to 10 attributes 	reduced to 29 attributes 	reduced to 3, added to the data. 

Applying GMM, then Neural Net using the Breast Cancer dataset:

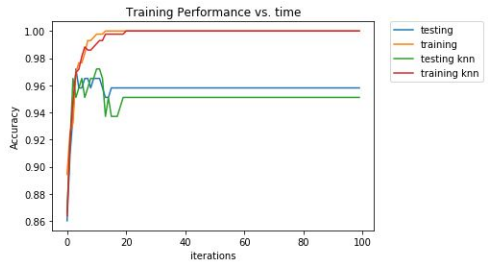
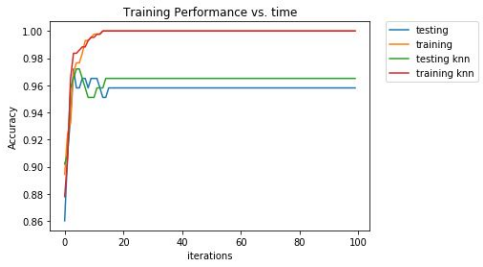
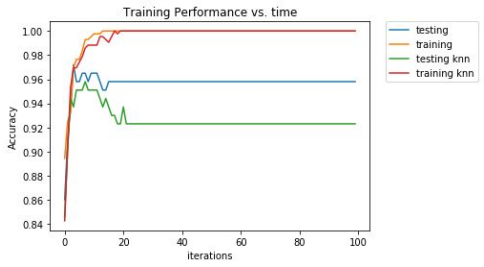
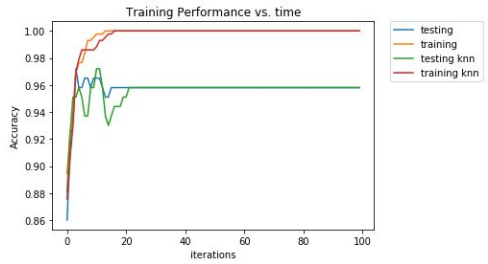
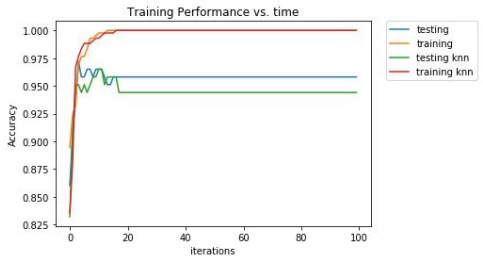
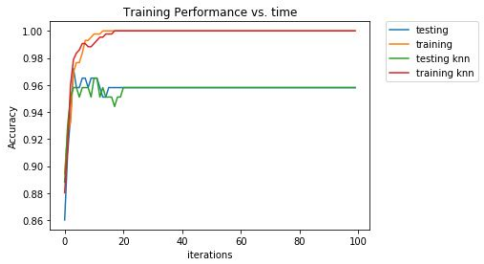
When the number of components was 100, the neural net given the extra GMM features performed slightly better on the validation set. There was a slight information gain however because the accuracy was so high even without doing GMM, there was little to improve. Overall I think GMM would definitely augment other NP-hard problems in classification with the help of GMM. The optimal level of components was probably near 100 or 200 since the performance with GMM was higher than without. The extra components were able to capture a few of the outliers as seen in the 3-D PCA model of the Breast cancer dataset. For the most part, just by looking at the cluster classification arrays, the clusters only changed slightly after applying GMM.

# of components = 2	# of components =10, GMM	# of components = 50, GMM
		
# of components = 100, GMM	# of components = 200, GMM	# of components = 426, GMM
		

Applying PCA then K-means on Breast Cancer dataset:

Overall adding PCA as an attribute to the Breast Cancer training data didn't improve the score compared to the regular training data. There was a slight increase in performance when the amount of clusters was 6. This was most likely due to how the neural net was fed in the classification. The Neural Net most likely neglected the added k means feature as k increased as each cluster's number i.e 1,2,3 etc.. are not linearly or mathematically related. The neural net was probably able to pick out a few features when the cluster number was low however once the number of clusters increased, the less related the information was. For example the neural net would have no idea what a cluster value of 60 vs 61 meant, it would try to figure out a linear or mathematical relationship between the numbers and the final classification when none exists. With a lower number like 6, there can be some information as there could be a "relationship" between the numbers 1,...6 and the classification and can tease out some mathematical

relationship. For the most part after applying PCA, each training element was in the same clusters as when PCA wasn't applied, showing that PCA altered the underlying clusters slightly.

Kmeans = 2	Kmeans = 6	Kmeans = 10
		
Kmeans = 20	Kmeans = 50	Kmeans = 200
		

Conclusion:

In conclusion, K means, GMM, and PCA are powerful tools for machine learning. Unsupervised learning on it own can be useful in visualizing data in 2D or 3D and can reduce the overall complexity of a problem. These tools can also help in improving the performance of Supervised learning problems through their use as an alternative "helper" attribute. Although the extent of this key insight was limited due to the simplicity of my training data, I know that unsupervised learning can greatly augment the performance of my supervised learning algorithms such as a neural network. PCA can be used to reduce the dimensionality and therefore the training time of a neural network whereas GMM and K means can be used to create a new attribute by clustering a training instance.

Citations:

[1] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
[2] https://www.kaggle.com/lucidlenn/sloan-digital-sky-survey
[3] https://scikit-learn.org/stable/modules/mixture.html