Assignment 3 – CS 4641 Keertana Subramani

Unsupervised Learning

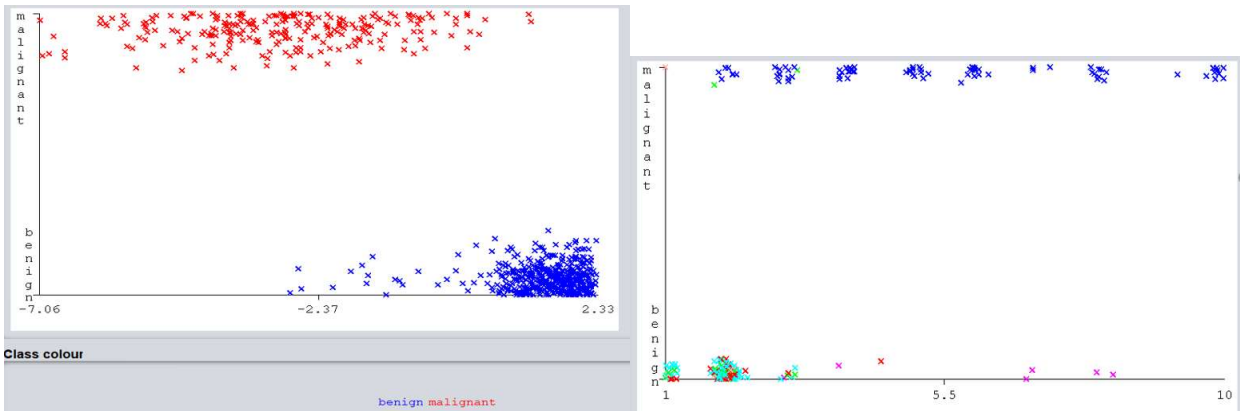
I. Description:

Dataset 1: Breast Cancer Dataset: This dataset, also used in Assignment 1, uses a number of features in a breast-cell to predict if a tumor is benign or malignant. The data has 699 instances of classified tumors and 9 related attributes, giving a sum total of 10 attributes, including the class of the tumor.

Dataset 2: Abalone Dataset: The Abalone dataset contains physical measurements of abalone used to predict their age. It contains 8 attributes as well as the number of rings in Abalone (which needs to be predicted by classifiers).

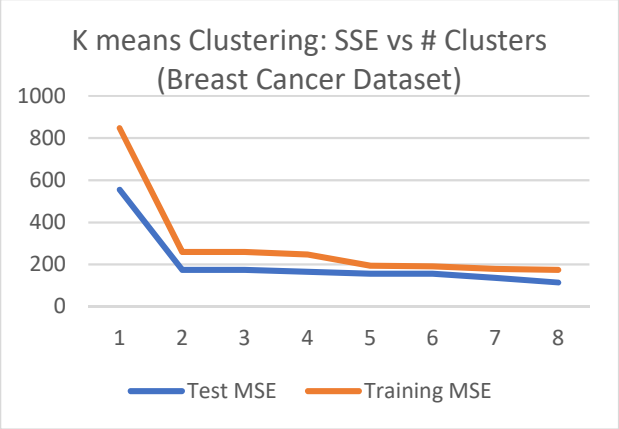
II. Clustering (Unsupervised Learning) Algorithms

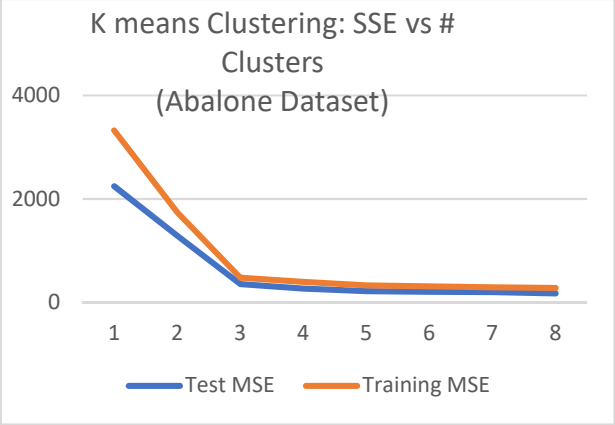
K-means: We use the simple K-means clustering in Weka, vary the number of clusters and study the resulting within cluster sum of squared Euclidean distances on a 66% split of training-test data. In this clustering algorithm, the parameter we can optimize is the number of clusters in the output. The k-means algorithm looks to minimize the inertia i.e., the sum of squares distances which indicate how close the clustered data points are to each other by taking the average of the squared distance of each data point to the cluster's centroid. In other words, the inertia indicates how coherent each cluster is. Given a constant number of clusters, The closer they are, the more robust the clustering is, and the better it will generalize to new, unseen test data. Using the elbow method to identify the optimal number of clusters to minimize the test and training sum squared errors, we see that 2 clusters leads to a significant decrease in within-cluster Euclidean-distances, with a further increase in number of clusters leading only a marginal decrease in this parameter. Even though more clusters decreases the sum of squared errors, it might result in overfitting the training set and fail to generalize well. With regards to the breast cancer dataset, where there are only two output classes (benign and malignant), it makes sense that the optimal number of clusters is 2. Moreover, as shown below, these clusters line up with the labels.



For the Abalone dataset, we see a similar drastic decrease in within-cluster sum of squared Euclidean distances, but the optimal number of clusters using the Elbow method is found to be 3. Increasing the number of clusters beyond this point does decrease the sum of squared errors, albeit at a slower rate, but may also overfit to training data.

A general concern with the K means algorithm is that it may get stuck at a local optimum due to its iterative nature and the random initialization of centroids at the start of the algorithm. To do this, given more time, it would be useful to run the algorithm with different initializations and observe results.



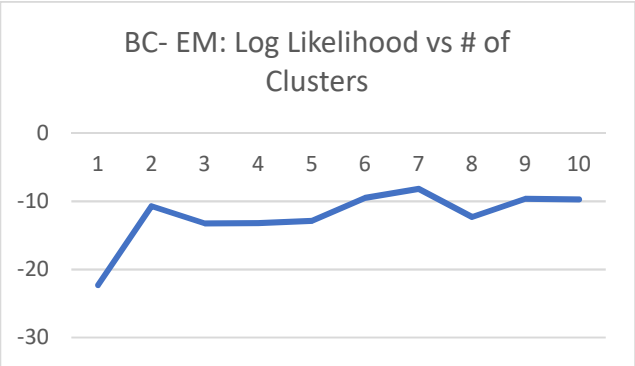


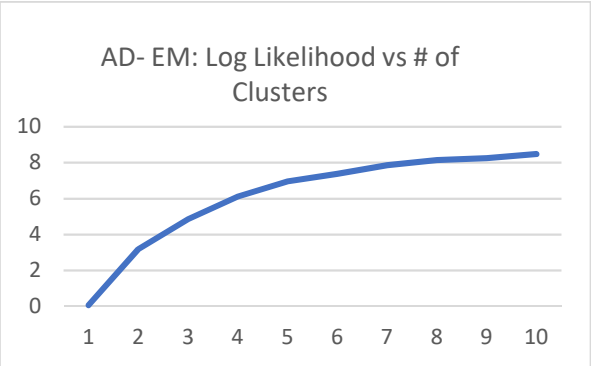
Expectation Maximization: EM is a soft-clustering algorithm that assigns the probability of a point belonging to any centroid instead of assigning definitive memberships of points to clusters. In fact, K means clustering is a special type of the Expectation maximization in which the points are biased or hard assigned to specific clusters. EM assumes that all data points are sampled from Gaussian distributions.

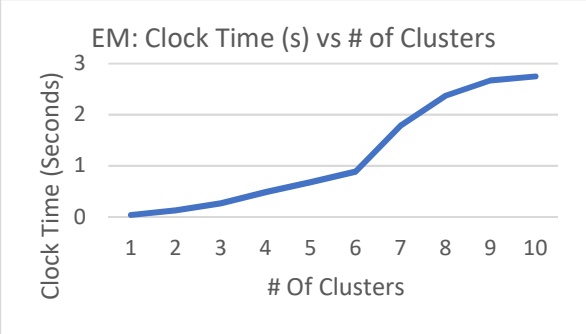
EM starts with a random set of components and modifies parameters so as to maximize the likelihood that the observed datapoint is generated by the Gaussian. It does this by computing the membership probability of each database record in each cluster and updating the mixed model parameters. A good way to observe the performance of the EM algorithm is to measure the log of the likelihood of the data points being generated by the given Gaussian. The higher this log likelihood, the higher the probability of the Gaussian generating the given datapoint.

The results from running the EM algorithm on the breast cancer dataset suggests a similar optimal number of clusters as the K-means clustering i.e., two clusters. Increasing the number of clusters may increase the log likelihood of generating the observed data, but has two issues: a) it is also likelier to overfit and b) it takes a much longer time to produce results. The second graph which plots the clock time vs. the number of clusters explains the rapid increase in algorithm runtime after n=6 clusters. This can be because the algorithm has to calculate the probability of every data point belonging to each of the 6 possible clusters and then assign it to the cluster which assigns it the highest consequential probability.

While the log likelihood shows a fluctuating increase in the breast cancer dataset with an increasing number of clusters, in case of the Abalone dataset, there is a steady increase in the likelihood that the Gaussian model will generate the particular data point. However, once again, to prevent possible overfit, it is useful to pick an optimal number of clusters – say around 3 or 4.







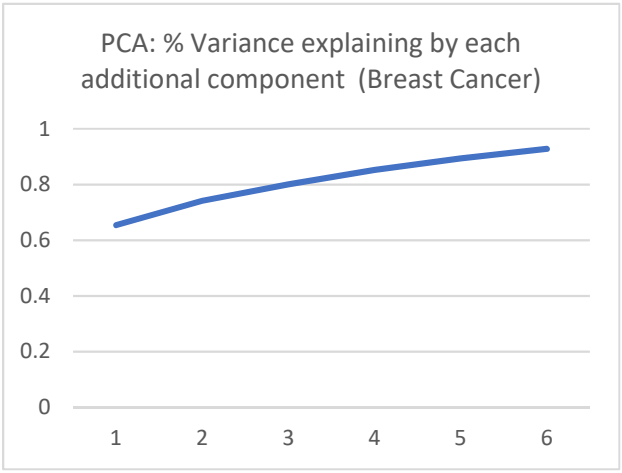
Both K means and EM will converge, but may or may not reach the global optimum since their optimal results depend on the initialization values. This is why it is useful to re-initialize both algorithms with different centers and observe changes in optima.

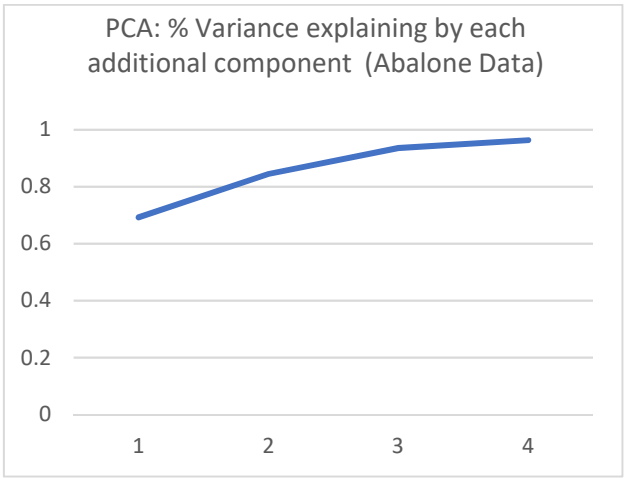
III. Applying PCA To Datasets:

Principal Component Analysis is a dimensionality reduction algorithm. PCA finds the directions of maximum variance in high-dimensional data and projects it into a subspace with fewer dimensions while retaining maximum possible information. We use the pre-processor tool in Weka and apply the Principal Components filter to reduce the number of dimensions in the dataset so as to define it using components that linearly combine features in such a way that they are able to describe the original dataset with the least dimensions while preserving maximum possible information. PCA, by reducing the number of dimensions for clustering, also has potential to increase runtime and decrease sum of squared Euclidean distances, in for instance, k means clustering. Reducing dimensionality also makes sense when there is a high correlation amongst variables, which points to potential redundant information in the original dataset.

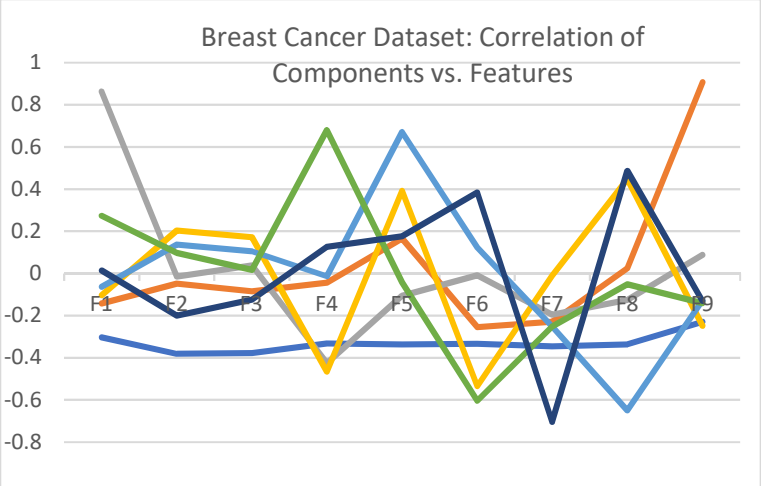
PCA tries to find components that explain the maximum possible variance in the dataset, ordered by the component that explains the most variance to that which explains least.

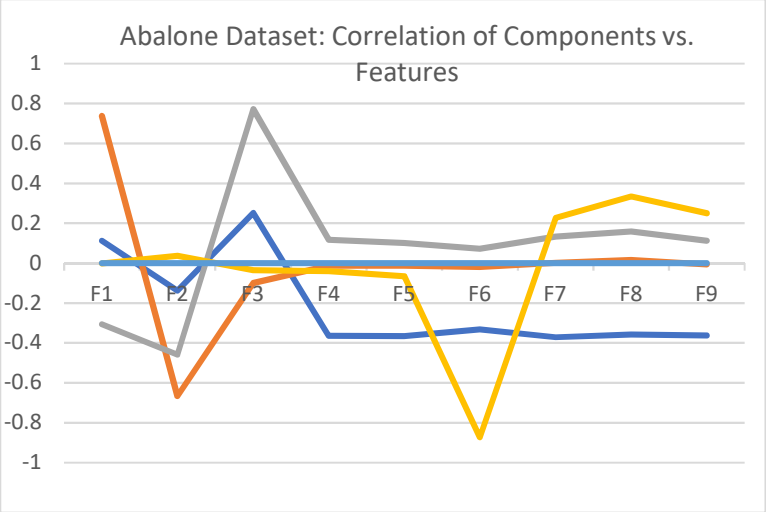
Components are sorted by their contribution to explaining variance, and in both cases, we see that the first component explains most (>50%) of the variance in the datasets with each subsequent component only marginally improving the proportion of original information accounted for.





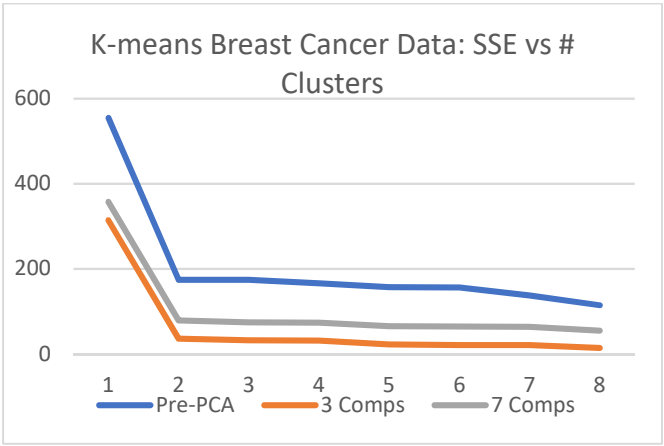
Below we see the correlation between components and features in the breast cancer and abalone datasets.

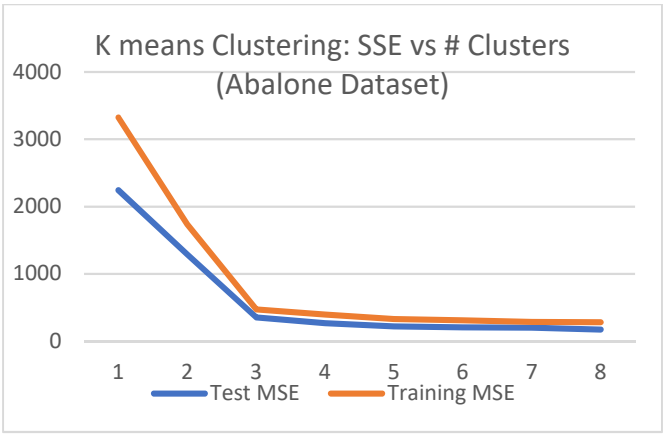




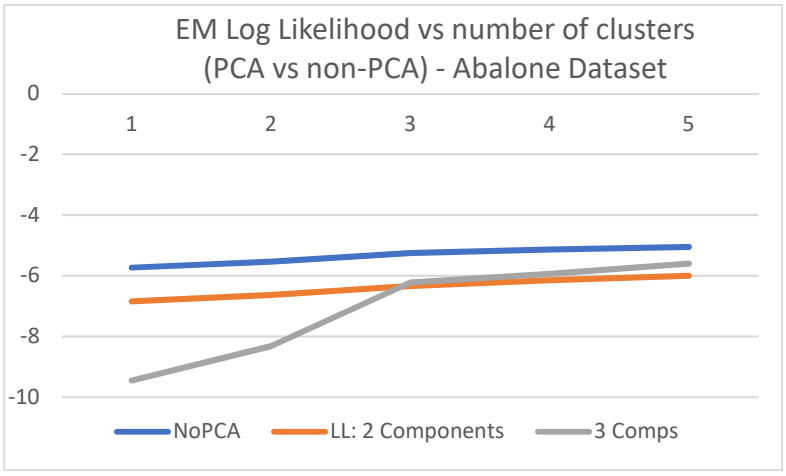
The higher the correlation between the components and features, the greater the proportion of the feature explain by the selected component. When we apply PCA, we want to select those components that indicate the information explained by the maximum amount of features in the dataset i.e., those components with high correlations to features.

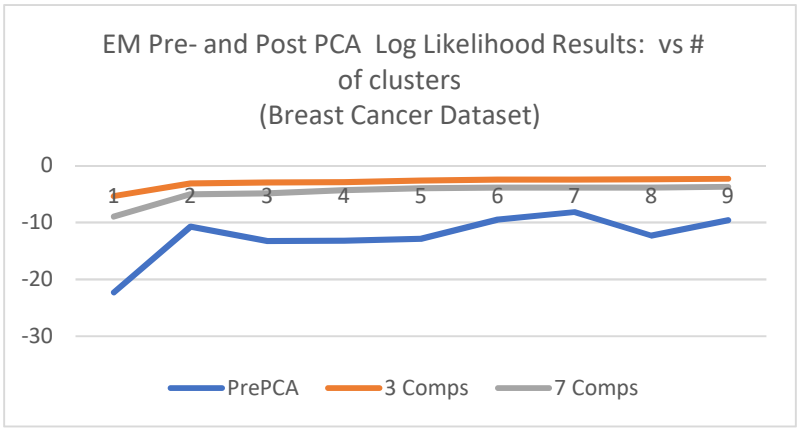
Clustering after PCA: We apply the k-means and Expectation Maximization algorithms on the dataset again after dimensionality reduction using PCA





We see that inertia decreases after applying the PCA dimensionality reduction to the original breast cancer dataset. This could be because, initially, we were in a space with very high-dimensions (a large number of features in original Breast cancer data), and since the performance metric is minimizing the squared Euclidean distances , it can become very inflated (also called a “curse of dimensionality”). Running a dimensionality reduction algorithm such as PCA prior to k-means clustering helps alleviate this problem and speed up the computations, as we see in the graph above.





After the PCA, when applying the EM algorithm, the log likelihood of the data point being predicted by the Gaussian decreases, implying that the algorithm is indeed more robust with the original dataset. This could be because applying PCA and reducing the number of components to retain only the most important information causes loss in some useful information that reduces the likelihood of datapoints being drawn from the given clusters. Under a low number of clusters, the resulting log likelihoods are much higher in the dataset with just 2 components rather than the one with 3,

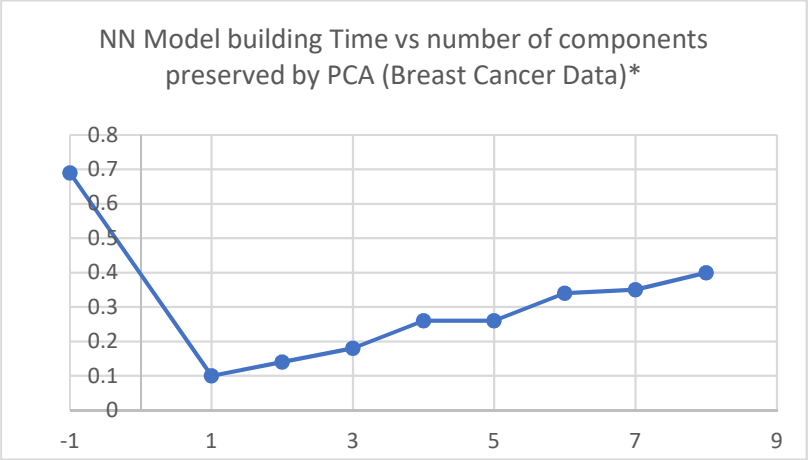
although the 3 component algorithm soon overtakes in performance as the number of clusters increases. This could be because the third component contributes only marginally to explaining additional variance in the Abalone dataset.

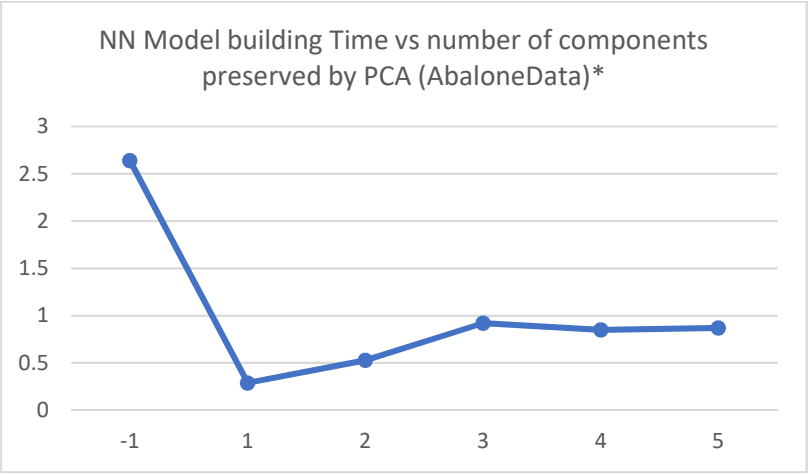
The differing performance of EM and K-means algorithms before and after applying PCA seems to suggest that the clusters formed before and after dimensionality reduction are different. This is because the centroids shift when data is projected from many dimensions onto some, thereby shifting cluster locations and densities.

III. Neural Networks:

We apply the neural network algorithms to both datasets before and after applying PCA and study the difference in results – in terms of both clock time as well as performance i.e., predictive accuracy of the neural network on test sets. Neural networks learn by taking in inputs, which are modified by an equation of sorts which changes their original value. The network then combines such weighted inputs with reference to thresholds and activation functions to give a result.

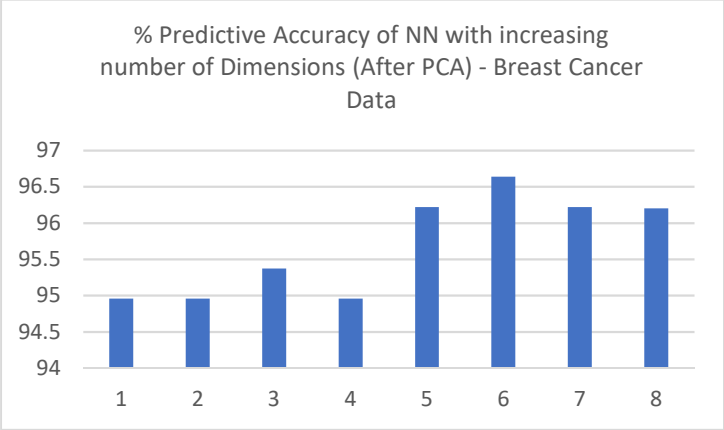
In the graphs below, the component index -1 corresponds to running the neural network before filtering the dataset through PCA.





Reducing the number of dimensions using PCA makes the Neural network build a robust model run much faster than when it has to consider all features from the original dataset. This is because the original data may contain redundant information, highly correlated information etc. which don't contribute much to making a sound, generalizable model. However, with dimensionality reduction, we are able to preserve the most important and relevant information while at the same time simplifying and reducing the amount of input data fed into the neural network to be linearly combined, networked and pruned using backprop.

For both Breast Cancer and Abalone datasets, the time taken to build a neural network model is highest before PCA is performance and lowest when PCA retains just the one top component that preserves most important information about the data I.e, the component that explains most of the variance in the distribution.

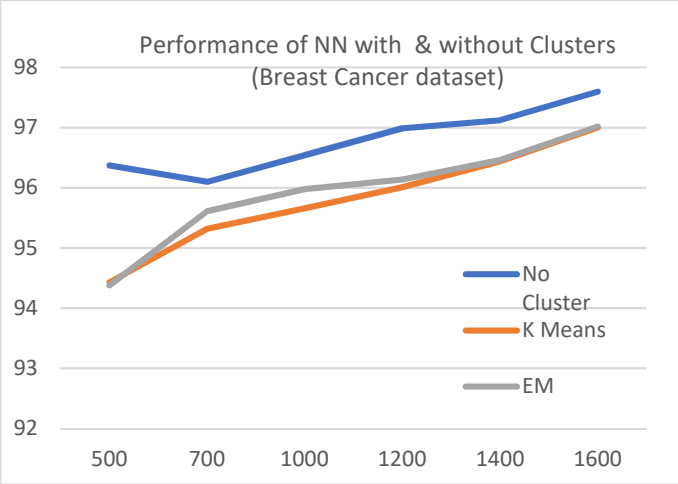


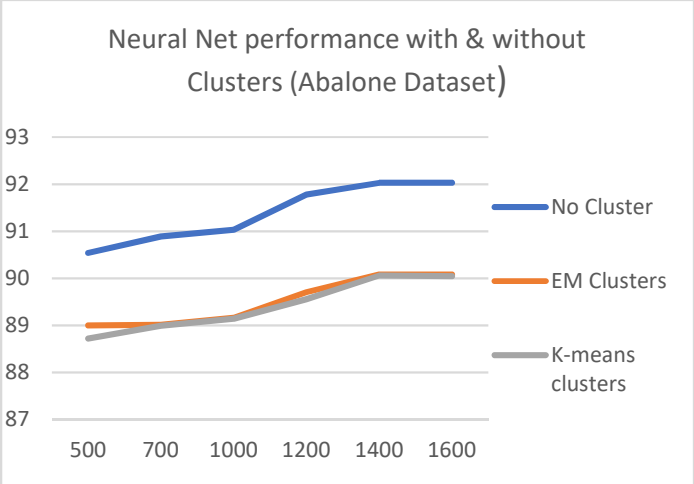
Predictive accuracy of the neural network increases as we increase the number of components preserved by PCA, reaches a peak and then begins to decrease as the model begins to over-fit to the training data, i.e., fit over noise and highly redundant (correlated) information. Predictive performance on test set peaks at n = 6 dimensions preserved by PCA for the breast cancer data. This is because, using the top 6 components produced by the PCA dimension reduction, we can explain the most important information about the data sufficient to build a sound neural network model using backprop that generalizes well without overfitting the data (as can happen if we increase the number of dimensions further beyond 6, which in fact, decreases predictive accuracy). We see a similar trend of increase in predictive accuracy with increasing preserved dimensions, followed by a decrease, in the Abalone dataset as well.

When Clusters are introduced as features: The training time increases significantly as number of clusters increases, but on the whole training time for neural networks with clusters is less than the training time for the entire original dataset. Increasing the number of training time on the neural networks, does increase accuracy, however, it increases at a decreasing rate. Amongst the two implementations of neural networks with, i.e., with EM clusters, K- there is not a substantial difference in performance. However, as compared to running the neural networks with original features, the performance declines considerably for neural nets implemented on clusters as features, because of loss of some useful information to generate a robust model.

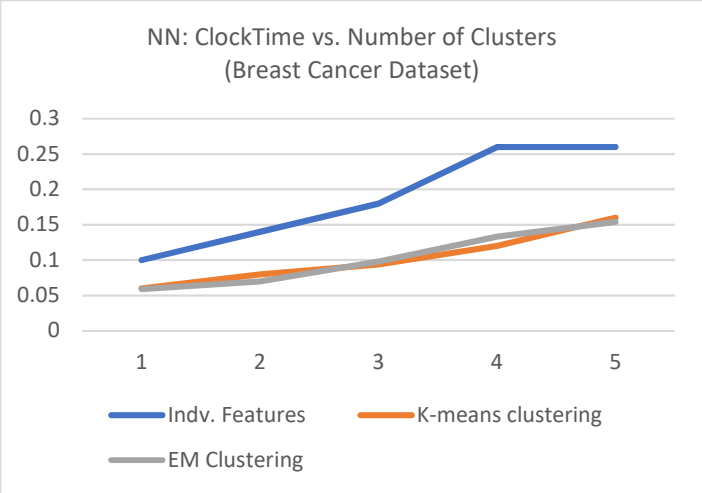
Even while the accuracy decreased significantly on clustered data, the speed to run the neural network was found to be much higher on clustered data than original data.

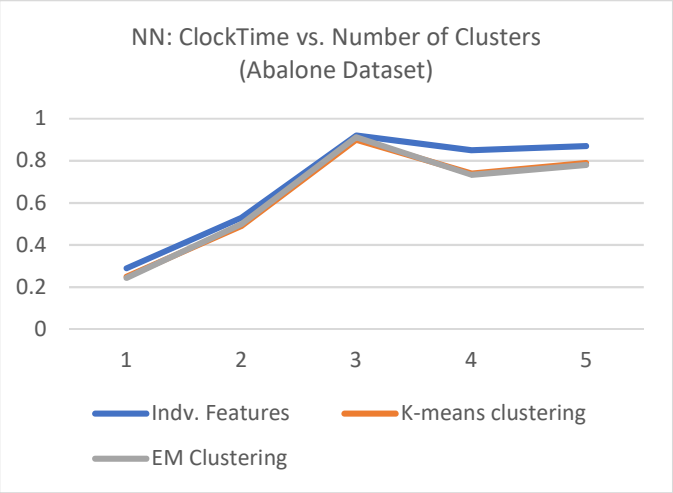
Neural Network Performance: With and Without Clustering





Neural Network Speed: With and Without Clustering





References:

1. The EM Algorithm for Gaussian Mixtures, https://www.ics.uci.edu/~smyth/courses/cs274/notes/EMnotes.pdf

2. https://stackoverflow.com/questions/24441924/principal-component-analysis-on-weka

3. https://futurism.com/how-do-artificial-neural-networks-learn