CS 4641 Assignment 3: Unsupervised Learning

Abstract – The project aims to explore unsupervised learning algorithms and their applications in different problems. The first part of the project was to analyze the performance of the k-means and GMMs via expectation maximization (EM) clustering algorithms. These algorithms were run on datasets used in the previous two projects on supervised learning and randomized search which were the mushroom and pulsar star datasets. The mushroom dataset contains a total of 8124 data points and 22 attributes which include color, shape, and the sizes of the different parts of the mushroom and a classification of whether the mushroom is edible or not. The pulsar dataset has 17898 data points with 8 attributes including standard deviation, mean, skewness of signal-to-noise ratio readings. Then, the principal component analysis (PCA) algorithm was used to reduce the dimensionality of the problem, and the K-means and EM were run again but on the new dataset formed by PCA. Lastly, neural networks was used to analyze the data with the PCA reduced dataset, the dataset with additional attribute of cluster information from the k-means algorithm, and the dataset with cluster information from the EM algorithm.

Introduction

Unsupervised learning is used in analyzing unlabeled and unclassified data. It identifies common features and reacts to new data by checking the new data's commonalities. In the previous project for supervised learning, the labels of whether a mushroom was edible and a star was pulsar were used in the learning algorithms. For this project, the algorithm will attempt to learn relationships between the data elements. Algorithms that will be used include the clustering algorithms: k-means and GMMs via expectation maximization (EM). Furthermore, a data reduction algorithm called the principal component analysis was used to see the effect of the performance of running the other algorithms on the reduced dataset. Waikato Environment for Knowledge Analysis (Weka), a machine learning software was used to run the algorithms. The following paragraphs will provide a short description about the algorithms.

K-means

The k-means clustering partitions n observations into k clusters in which each observation belongs to the cluster with the nearest mean. The algorithm receives an input of the number of clusters, k, and the set of points which will be the data points in this case. Next, k centroids are placed in random locations. Then, an iteration of assigning clusters to the points and setting new centroids based on the points in a cluster are calculated. This iteration is repeated until a convergence is reached. For the analyses, the sum of squared error (SSE) will be used with the formula as follows:

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x) \quad (1)$$

where x is a data point in cluster C_i and m_i is the representative point.

GMMs via expectation maximization

EM is an iterative algorithm that starts from some initial estimate, and then proceeds to iteratively update until convergence is detected. The convergence is detected by computing the log-likelihood after each iteration and halting when significant changes are not observed from one iteration to the next. The log likelihood measure will be seen in the graphs generated from running the algorithms and the formula for it is in the form:

$$\log l(\Theta) = \sum_{i=1}^{N} \log p(\underline{x}_i|\Theta) = \sum_{i=1}^{N} \left(\log \sum_{k=1}^{K} \alpha_k p_k(\underline{x}_i|z_k, \theta_k) \right) \quad (2)$$

Where p_k is the Gaussian density for the kth mixture component.

Principal component analysis

The PCA uses transformations to reduce data from n-dimensions to k-dimensions. In the reduction process, the covariance matrix is computed. From this matrix, the eigenvectors are calculated using the singular value decomposition. In this document, the datasets are compressed to 2-dimensions because it is easy to visualize the clustering of data in small dimension.

Discussion

Most of the default configurations in Weka were used. One parameter that was set constant was the seed value which was set to 10. The data was split into 70 % training and 30% testing set. The number of clusters was ranged from 1 to 15 in increments of 2 and the training time for neural networks was capped at 60 seconds for the sake of time management for this project.

The first experiment conducted was running the k-means algorithm on the two datasets and results can be shown in the figure below.

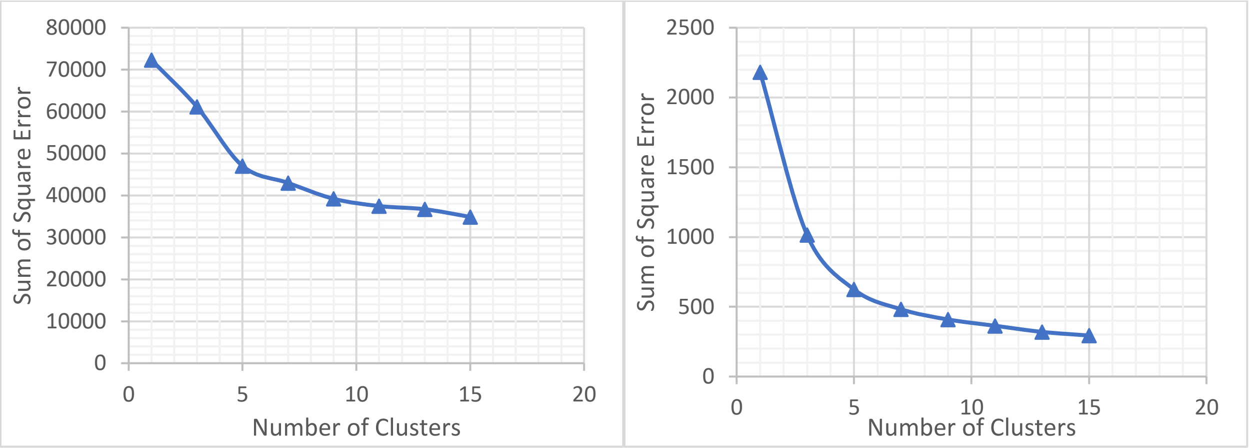


Figure 1: Effect of the number of clusters on the sum of square error for the mushroom (left) and the pulsar star (right) dataset

Figure 1 shows that the SSE is indirectly proportional to the number of clusters for both datasets. This was to be expected because the radius of clusters become smaller with more clusters and data points are able to find near cluster centroids which result to small SSE. The first dataset has a more linear relationship while the second has a steep drop in error in the lower numbers of clusters. This might be because of the difference in the number of attributes affecting the classification. Moreover, the smaller changes in errors at high numbers of clusters probably resonated with the fact that they are reaching the numbers equivalent to the number of attributes of the datasets.

The second experiment studied the performance of the EM algorithm on both datasets. Unlike the first experiment, the dependent variable values had a more similar scale because the result was in logarithmic scale.

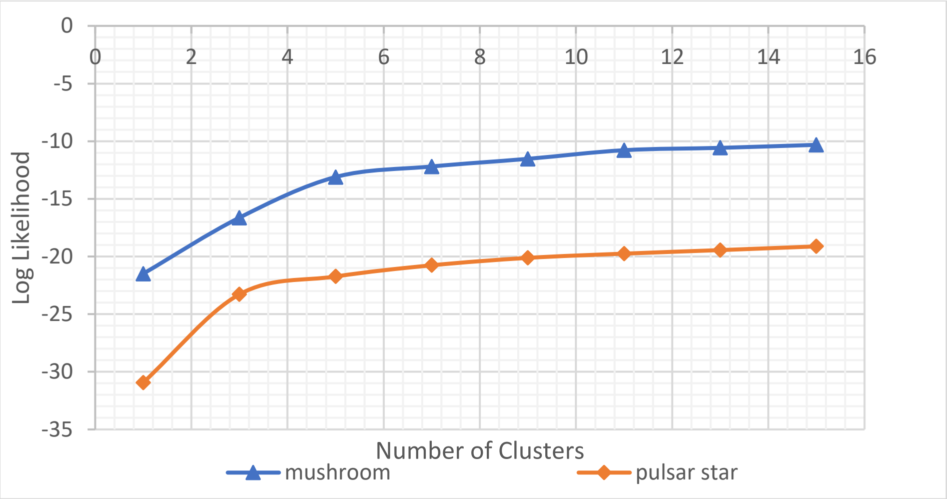


Figure 2: Effect of the number of clusters on the log likelihood for the mushroom and the pulsar star dataset

Figure 2 shows a similar relationship between the two datasets. Both showed a positive change with an increase in the number of clusters which is similar to that of the first experiment. Furthermore, like the first experiment, both datasets plateued or showed less improvements at higher number of clusters.

The third experiment uses the PCA algorithm. In the next four figures, images of data clustering will be shown and differences and similarities will be reviewed for each dataset.

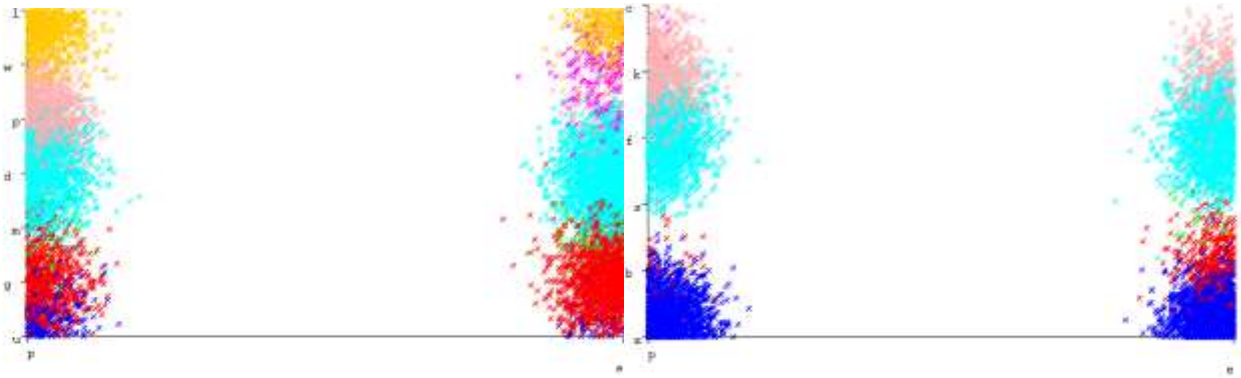


Figure 3: Raw mushroom data looking at habitat on the left and cap shape on the right with respect to their classification

The left side of Figure 3 shows that that there isn't a clear relationship of the attribute habitat for the classification of edible mushrooms. This can be inferred by looking at the amount of data in each color in the two sides: p (not edible) and e (edible). For most of the colors, the amount is split almost equally to the two sides which shows a randomness in the relationship. Same goes for the right side where the amounts of each color is also split almost equally into both sides.

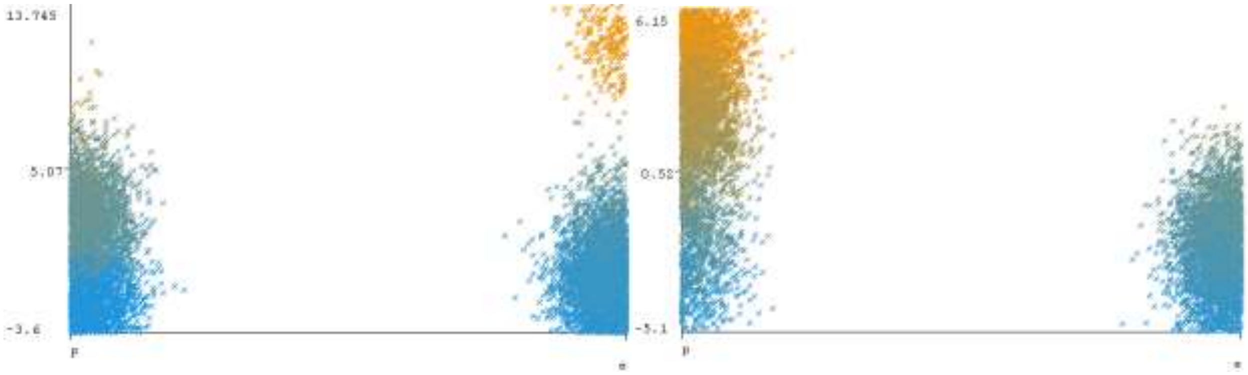


Figure 4: Mushroom data after PCA with 2 new attributes

Figure 4 shows a dissimilar result as that of Figure 3. It shows an unequal distribution of data, which is an evidence of the use of the dimensionality reduction algorithm. Although there is a significant change in the clustering in Figures 3 and 4, the other attribute clusterings had more similar clusterings as Figure 4, which contributed in the PCA algorithm. Furthermore, the general feature of the data shouldn't have been altered as the PCA promises.

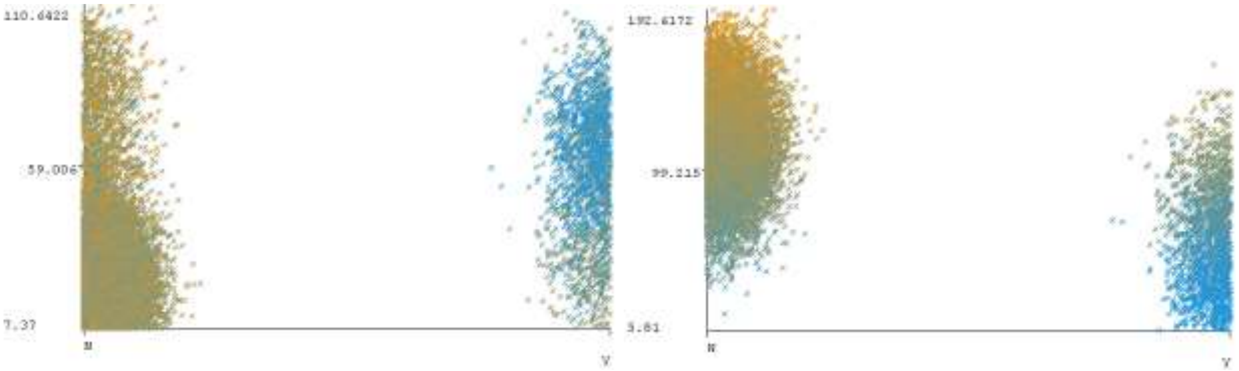


Figure 5: Raw pulsar star data looking at standard deviation of the DM-SNR curve on the left and mean of integrated profile on the right with respect to their classification

Figure 5 shows the distribution of data into the classification: N (not pulsar) and Y (pulsar). Unlike mushroom's dataset, there is more unequality in the distribution of data here.

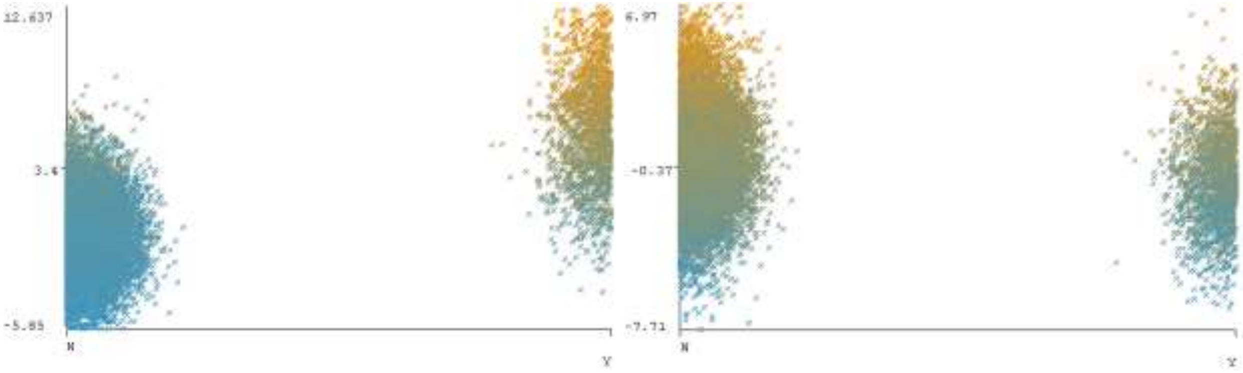


Figure 6: Pulsar star data after PCA with 2 new attributes

Figure 6 shows the result of the reduction algorithm on the pulsar star dataset. Unlike the mushroom dataset after PCA, the pulsar dataset seems to have a more combined data of attributes as seen in the different colors being in the same cluster. The difference might have risen from the differences in the eigen values as shown in Table 1 below.

Table 1. Eigen values for the new attributes obtained from performing PCA

Attribute	Eigen Value
Mushroom New Attribute 1	7.78
Mushroom New Attribute 2	9.84
Pulsar Star New Attribute 1	4.13
Pulsar Star New Attribute 2	2.14

The eigen value is directly related to the variance of the data. That means that mushroom has higher variance than pulsar data. This aligns with what we can see in Figure 4 and 6 as mushroom’s dataset is more spread out than that of pulsar star.

The fourth experiment runs the k-means algorithm to the reduced dataset, and the result is shown in the figure below.

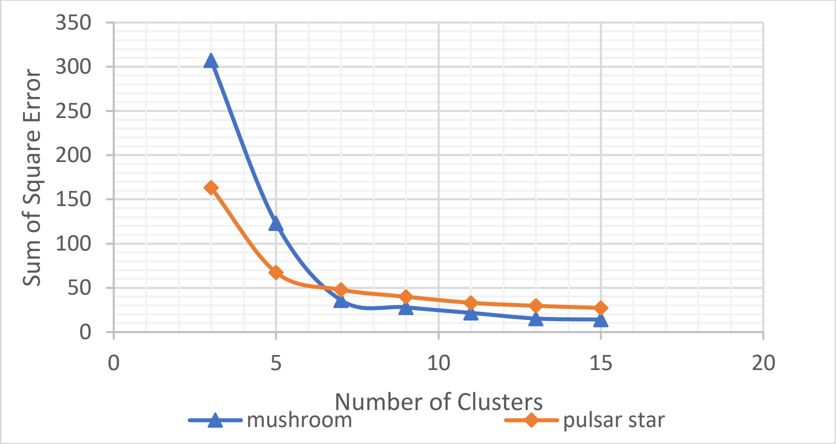


Figure 7: Effect of the number of clusters on the sum of square error for the mushroom and the pulsar star dataset after using PCA

Figure 7 shows a significant change in the scale of SSE (60,000 to 300 for mushroom and 2000 to 150 for pulsar star) looking at a value of about 3 in the x-axis. This is obvious as the number of attributes was broken down into just 2. Unlike Figure 1, the pulsar star dataset has a more linear relationship now which is the opposite of the analysis in Figure 1. However, even if the dimension is scaled down, the plateuing can still be observed at higher number of clusters. This shows the consistency of the data even after running the PCA algorith.

The fifth experiment runs the EM algorithm to the reduced dataset, and the result is shown in the figure below.

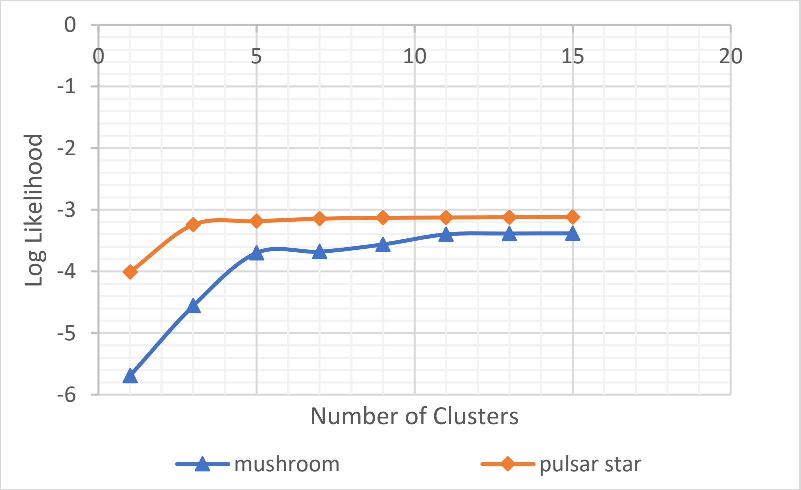
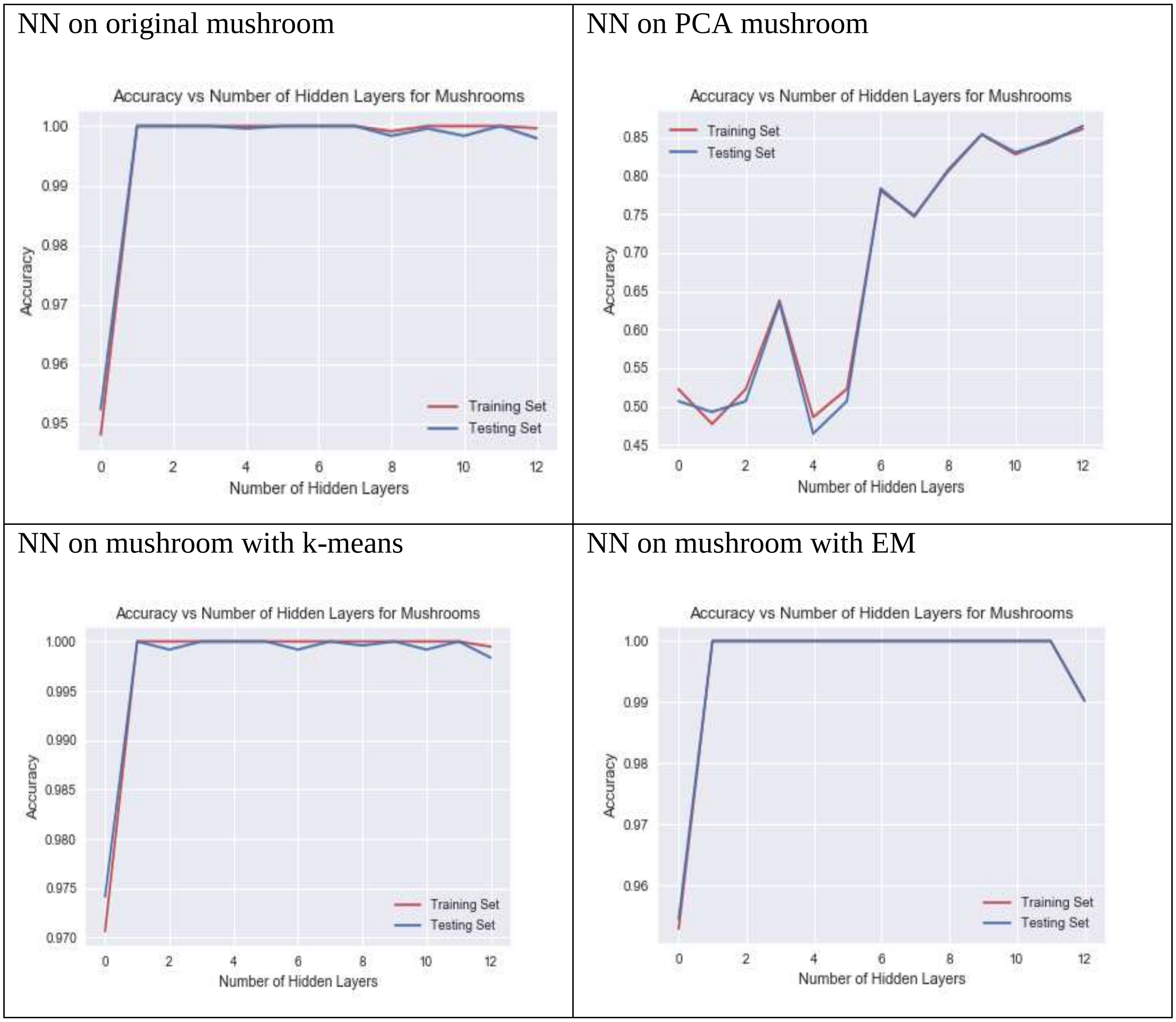


Figure 8: Effect of the number of clusters on the log likelihood for the mushroom and the pulsar star dataset after using PCA

Like Figure 7, the scale of the log likelihood was reduced. The order of top and bottom was also changed with pulsar star now at top of mushroom. This was also the case in Figure 7 which maybe is a result of the mathematical aftereffect of the PCA algorithm. And as stated previously, the steadying of the data is seen in the higher number of clusters.

The last experiments take either the original dataset, reduced dataset through PCA, dataset with k-means clustering data, or dataset with EM clustering data. The neural network code used in the previous project which implements with scikit-learn was used for these experiments.



NN on original mushroom

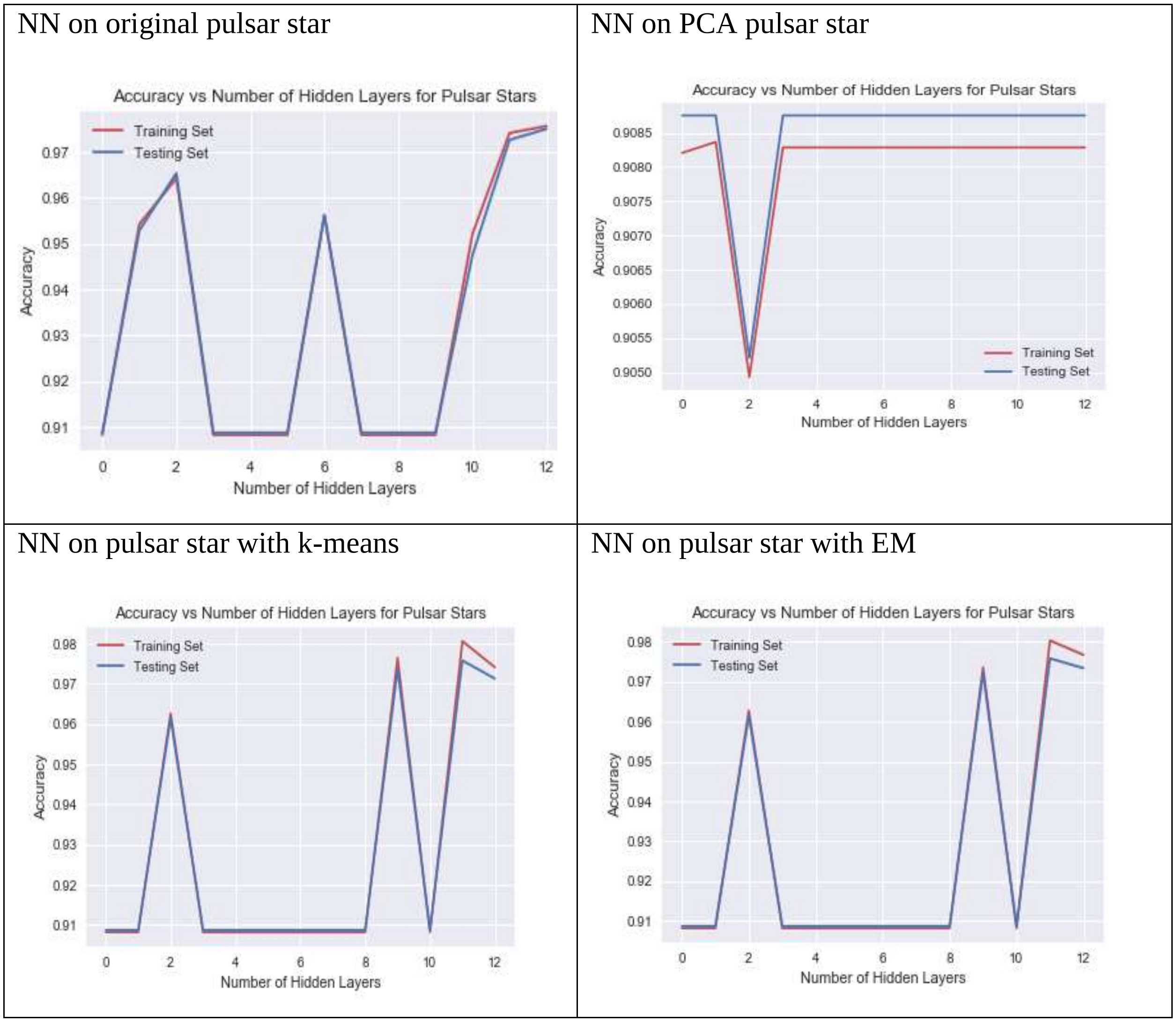
NN on PCA mushroom

NN on mushroom with k-means

NN on mushroom with EM

Figure 9: Neural network performance on variations of the mushroom dataset

From Figure 9, the performance of neural networks is similar for the original, with k-means, and with EM. With a small difference, EM has done a better job than k-means, but all the three implementations achieved a result above 99.5 %. On the other hand, there was a decrease in accuracy when PCA was used. This was to be expected because the PCA reduced the data into only 2 attributes which the neural networks will have to generalize and underfit the data. The trend seem to be that the accuracy is directly proportional to the number of hidden layers which is a common trend when using neural networks.



NN on original pulsar star

NN on PCA pulsar star

NN on pulsar star with k-means

NN on pulsar star with EM

Figure 10: Neural network performance on variations of the pulsar star dataset

Like Figure 9, Figure 10 also shows similar performance for the original, with k-means, and with EM datasets. The k-means and EM performed almost identically. Furthermore, the neural networks was able to perform well in all cases because of high accuracy. Unlike the reduced mushroom dataset, the reduced pulsar star dataset did not have an increasing trend which means that the trend shown in the recued mushroom set cannot be generalized.

Conclusion

Based on the experiments, this paper was able to validate the positive trend in increasing the number of clustering to the SSE and log likelihood. It was also able to show that the higher the eigen value, the data points are more spread out because of higher variance in the dimensionality reduction through PCA. However, the experiments were not able to obtain conclusive findings in the neural networks experiments as the accuracy levels were too close to each other. When running the algorithms in Weka, the k-means algorithms had shorter running times which is one of the advantages of the k-means algorithm. However, the accuracy for the dataset with k-means had a little lower accuracy than with EM. More experiments with other datasets should be conducted to further verify the theory behind the unsupervised learning algorithms.

Reference:

https://www.ics.uci.edu/~smyth/courses/cs274/notes/EMnotes.pdf

http://user.engineering.uiowa.edu/~ie_155/lecture/K-means.pdf

https://georgemdallas.wordpress.com/2013/10/30/principal-component-analysis-4-dummies-eigenvectors-eigenvalues-and-dimension-reduction/

https://hlab.stanford.edu/brian/error_sum_of_squares.html