Rohith Krishnan
CS4641

Unsupervised Learning Analysis

Datasets

Student-Scores. This data set came from a survey of secondary school students in Math and Portuguese courses and has information on a student's gender, family background, alcohol consumption habits and grades at three different times during the semester. The scores in the dataset were on a numeric scale from 0 to 20 and I modified the dataset to have only four classes for the final score where each class represented a range of scores on the original grading scale (i.e. grades in the range from 0 to 4 were mapped to a '1', ranges from 5 to 10 were mapped to a '2' etc.). The classification problem was to predict the final course grade given the various social statistics of each student. This dataset contains 32 features and 1044 samples.

Wine Quality. This dataset contains physiochemical data about different types of Portuguese white wine and contains a quality score of the wine sample from 0 to 10. The classification problem was to predict if a wine sample had a quality score above 5 or below 5 given different data such as acidity and pH levels. The data has 12 features and 4899 instances.

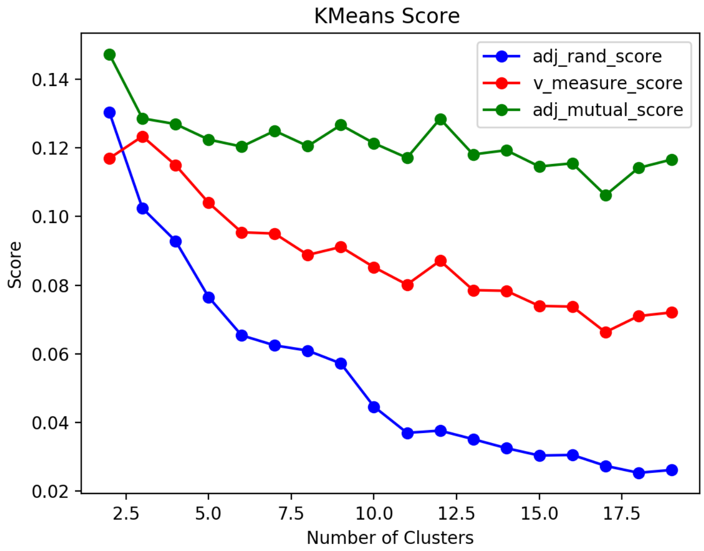
Why These Datasets are Interesting

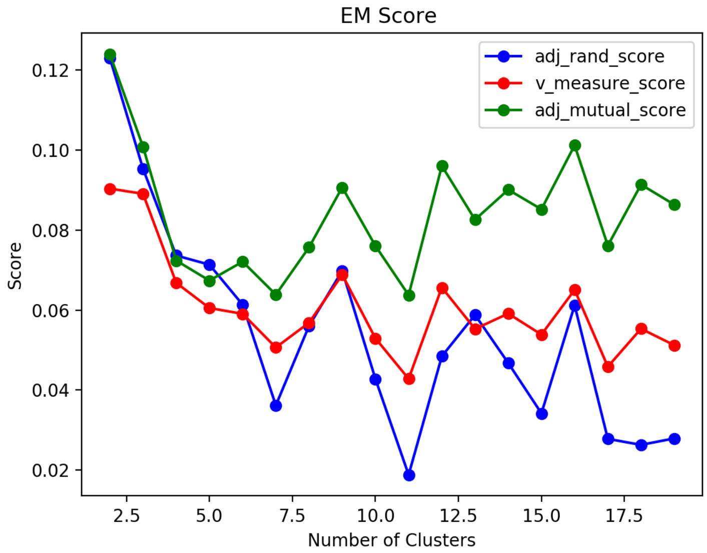
Before applying the different clustering and dimensionality reduction algorithms, I found the Student dataset interesting for the social implications that come from it. With this dataset, one could train different machine learning models to determine how likely a student is to fail a course given their social background and alcohol habits. These insights can give universities a better idea of what external factors contribute to student performance and what resources need to be made available to students to overcome these factors. Since the dataset is labelled, I was interested to see how the different clustering algorithms would cluster the data and if they could make clusters were all the members of a single cluster had the same label. Also, since the dataset has many features, I wanted to see how using dimensionality reduction on this dataset could help a neural network's accuracy in predicting the final score of the students in this dataset. As for the wine dataset, using clustering algorithms can help determine how similar two samples of wine are to each other. This kind of insight can help avid wine enthusiasts and collectors determine similarities and differences between different wine types.

Clustering

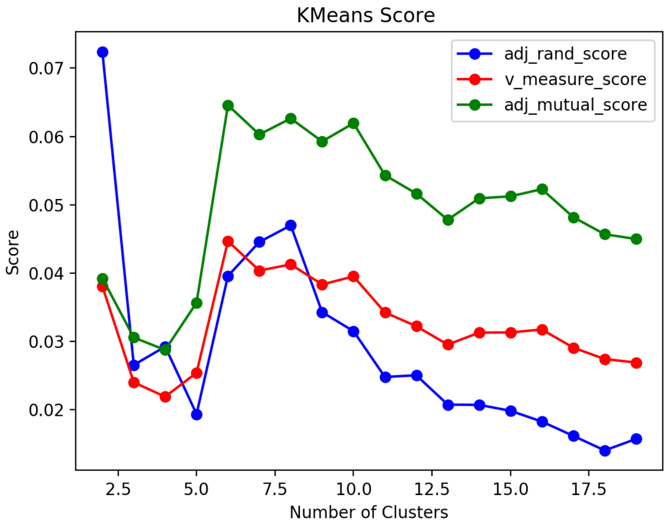
I first ran the k-Means and Expectation Maximization clustering algorithms from scikit-learn on my datasets. The scikit implementations of these algorithms use Euclidean distance to calculate intra-cluster distances between points. I ran both algorithms on different values of k from k = 2 to k = 20 and recorded the Adjusted Random Index (which measures the similarity of cluster-assigned labels and true labels), Adjusted Mutual Info Score (measures agreement of cluster-assigned labels and true labels), and the V-Measure Score (summarizes how well the algorithm makes clusters that only contain members of one class and that all members of a given class are assigned to one cluster). For the neural network experiments discussed later, I choose the k that had the highest V-Measure Score.

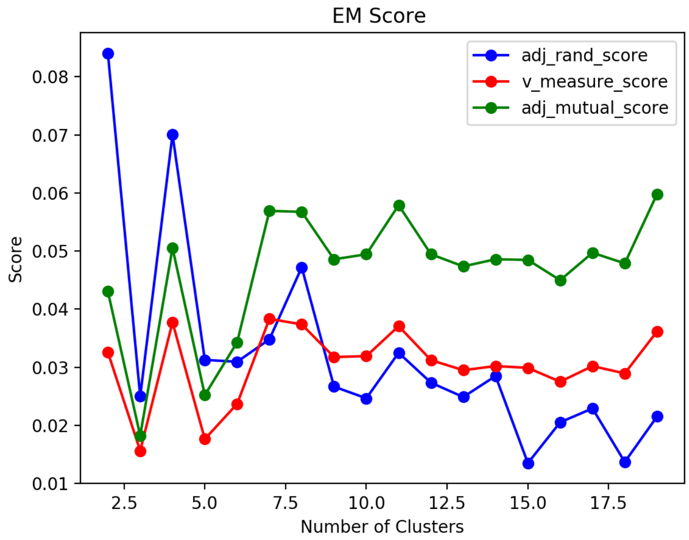
Student Dataset





Wine Dataset





For both datasets, the two clustering algorithms did not perform that well and seemed to struggle to make clusters that lined up with the true labels as evident from the low scores across the three metrics. For the student dataset, the scores across the three metrics were at their highest when k = 2, which means that two clusters were better at separating the data along the different classes in the dataset. This could be caused by the fact that most of the samples in the student dataset were in two out of the four possible classes which would describe why two clusters were the best at separating the dataset into homogenous groups. Both algorithms performed worse on the wine dataset when compared to the student dataset which could be due to the more complex distribution of the wine data.

Another interesting note is that Expectation Maximization had a lower V-score than K-means on both datasets. This result comes from the inherent nature of EM. While K-means tries to find distinct clusters by calculating the closest centroid for each sample, expectation maximization is calculating the probability that a sample belongs to a certain cluster out of the k clusters that are desired. This results in the algorithm finding soft-clusters which would allow a sample to belong to two clusters at the same time based on the probability distribution used in the algorithm (in this case scikit-learn uses a Gaussian Distribution). This property would result in clusters that are not as homogenous or complete as clusters found by K-means.

The poor performance in general could be attributed to the distance function that was used by both algorithms. Because both datasets have a relatively large number of features compared to samples. Therefore, the Euclidean distances calculated in each algorithm would become inflated and would result in the inaccurate clusters described by the metrics above. One possible change could be a distance function that better captures the domain knowledge needed for each dataset as Euclidean distance seemed to be too general for these datasets. For Expectation Maximization, a Gaussian Distribution may not accurately describe the underlying distribution of each dataset. Since the algorithm can use any probability distribution, we could find a distribution that accurately matches the underlying distribution of each dataset, which would result in clusters that more closely line up with the true labels of the data.

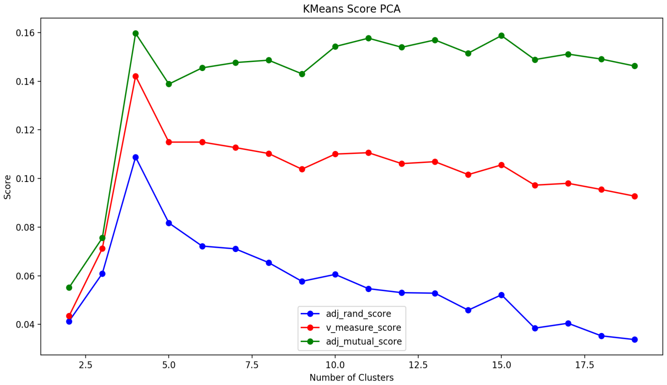
Dimensionality Reduction

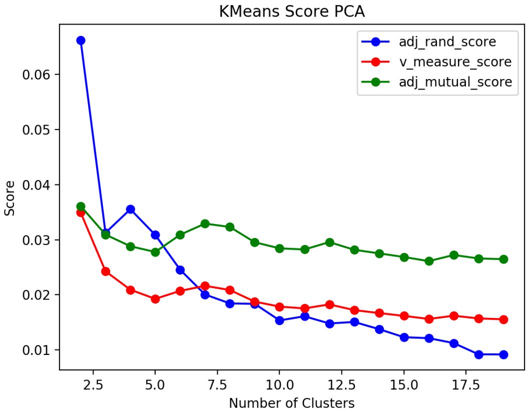
Both datasets seem to be impacted by the curse of dimensionality as they both have many features compared to the number of samples. Using dimensionality reduction algorithms would combine or remove redundant or irrelevant features and could improve the clusters with respect to the class labels in each dataset. I ran four different dimensionality reduction algorithms on the two datasets and re-ran my clustering experiments on the reduced datasets.

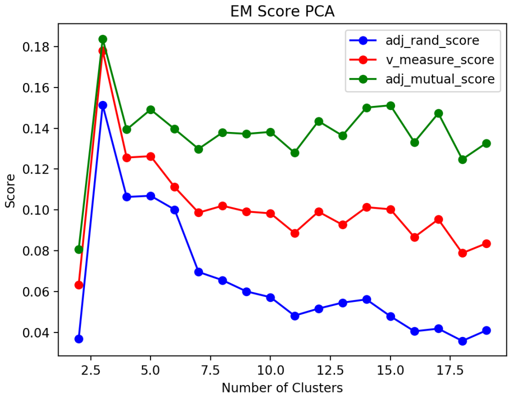
PCA

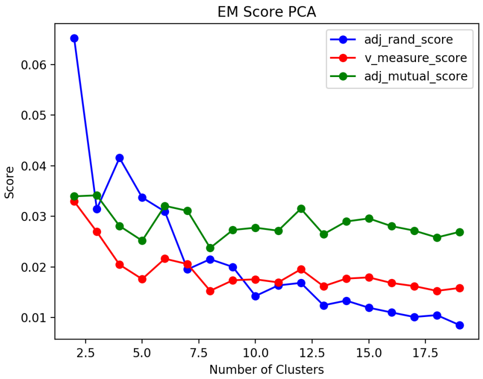
Student

Wine







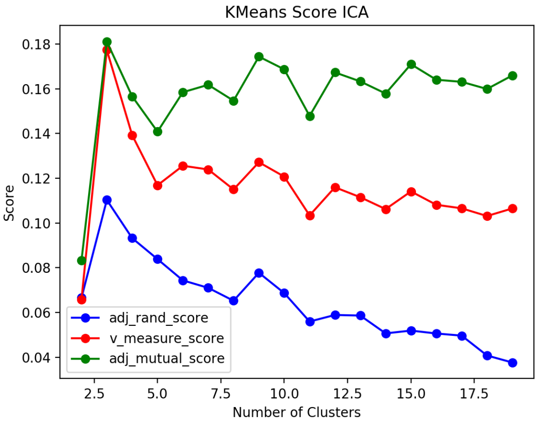


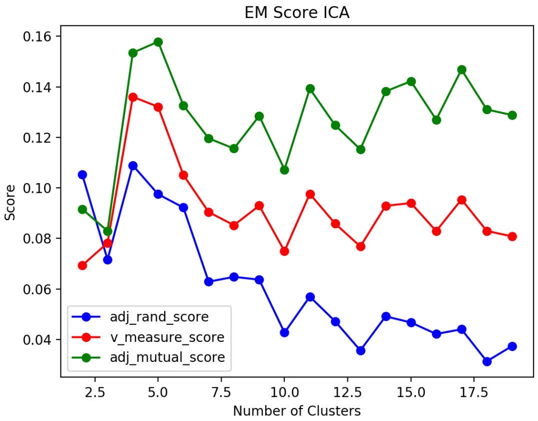
Dataset	Reconstruction Error	Eigenvalues
Student	0.719699817086	[3.39359075, 2.53791227, 1.7546735, 1.62925774]
Wine	0.56550184727	[3.27655388, 1.60692374]

For the student dataset, I ran PCA to reduce the dataset down to four components and I reduced the wine dataset to two components. My intuition was that since there are only four classes, four components could better reflect what class each sample belongs to and could help create better clusters for the two clustering algorithms. This same intuition was used on the wine dataset and for the other three dimensionality reduction algorithms. The reconstruction error is much higher for the student dataset than the wine dataset which could be caused by the large number of features that the dataset started with and by the fact that PCA creates linear combinations of less relevant or redundant features into a new feature space. This property would result in quite a lot of information loss for the student dataset when one tries to invert the transformation. Despite the high reconstruction error, PCA did help improve the performance of the clustering algorithms on the student dataset for as seen in the graphs above. On the other hand, the clustering algorithms performed worse on the PCA reduced wine dataset even though it had a lower reconstruction error and larger variance of its eigenvalues. It is possible that using only two principle components may have oversimplified the dataset and would have resulted in a feature space that was difficult for the clustering algorithms to separate on.

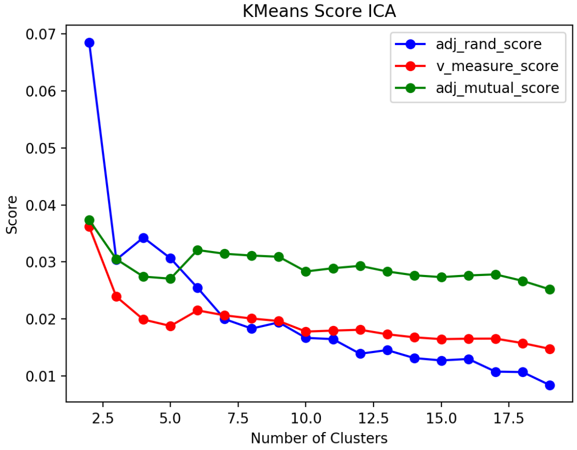
ICA

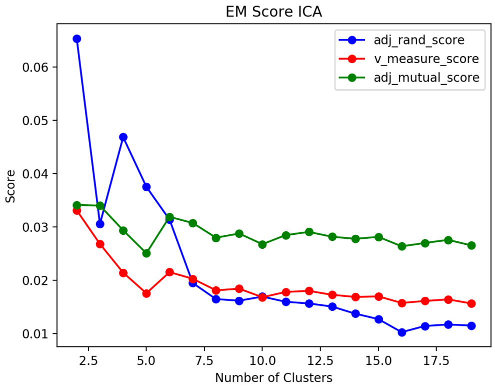
Student


KMeans Score ICA


EM Score ICA

Wine


KMeans Score ICA


EM Score ICA

Dataset	Kurtosis of Components
Student	3.72732283, 3.64470482, 2.75482768, 3.22483769
Wine	3.93087581, 2.18107462

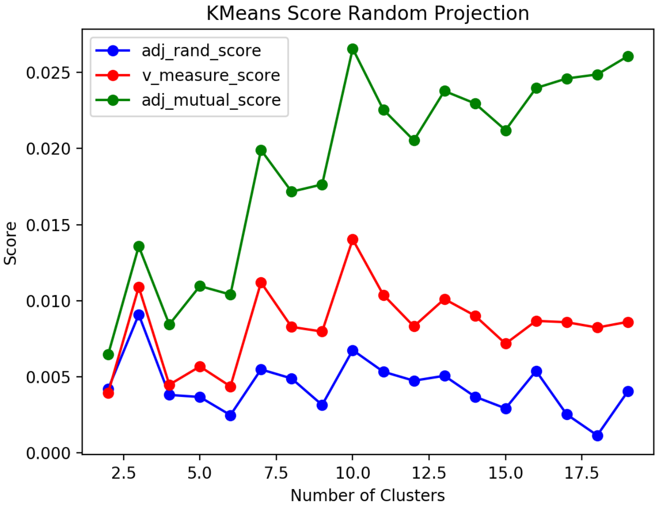
Running ICA on the student dataset resulted in similar performance when compared to PCA for both clustering algorithms although it did change the value of k where all three of the metrics peaked. I expected ICA to perform as well as or better than PCA as ICA tries to find the independent features that best describe the data. This notion of independence probably resulted in clusters that were more distinct which would explain the higher V-measure scores for K-means and EM. The wine dataset did not see any improvement in performance which could be explained by the kurtosis of the projected components. The kurtosis of the two components were further away from 3 than the components of the student dataset. This shows that the projected components are not Gaussian which was the same issue with the original dataset in terms of clustering performance.

Random Projections

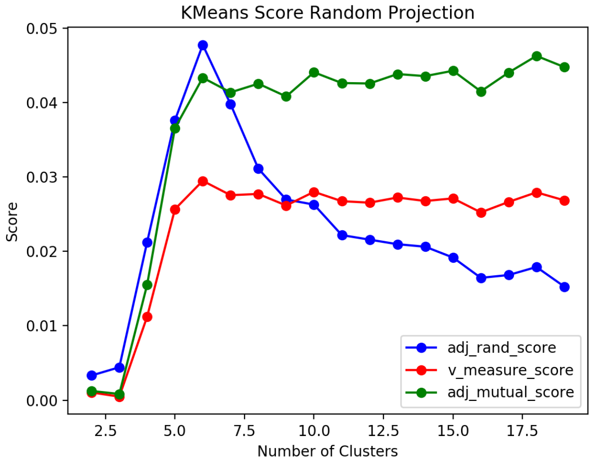
Using a random matrix to project the components into a smaller feature space heavily impacted the performance of the clustering algorithms. It also led to a large amount of information loss as evident by the high reconstruction error of both datasets. The student dataset saw a drastic change in clustering performance and had a higher reconstruction error than the wine dataset. The random projection used could have combined features that were not well correlated with each other and could result in some very odd cluster shapes that neither algorithm could handle well. Random Projections did cause some improvement for Expectation Maximization on the wine dataset. This could be the case that the projection got lucky and projected the wine data into a feature space that had more distinct clusters that EM was able to find.

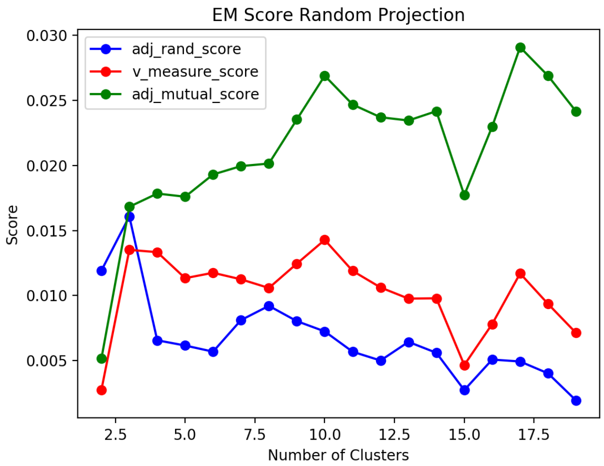
Dataset	Reconstruction Error
Student	0.880796845657
Wine	0.832276139341

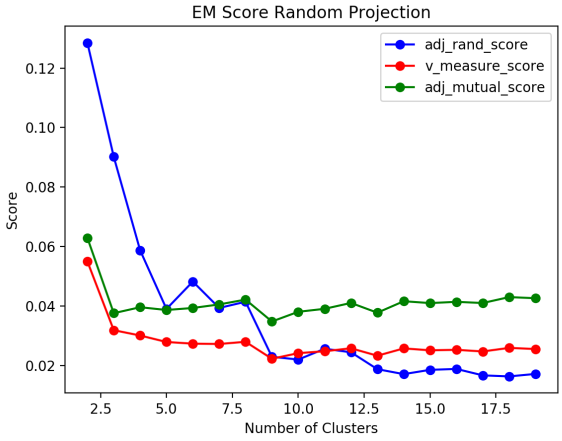
Student



Wine





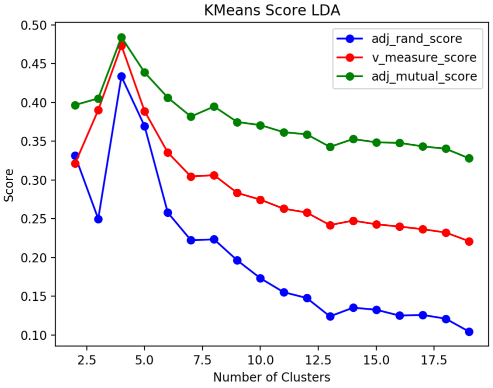


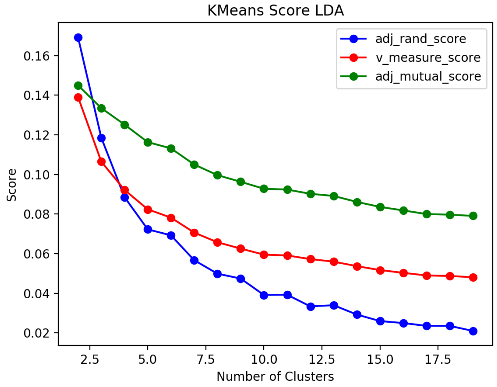
Linear Discriminant Analysis

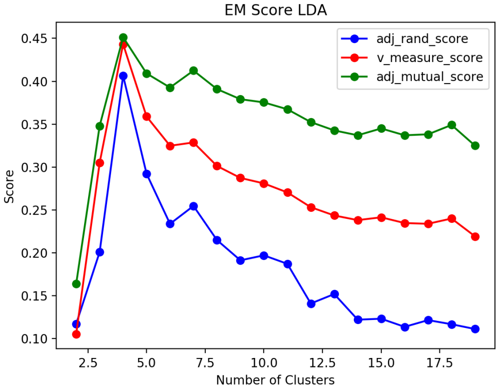
The final reduction algorithm I used was Linear Discriminant Analysis. This algorithm saw a large improvement in performance for the two clustering algorithms on both datasets. This is because this algorithm projects the input data into a linear space where the directions of the space maximize separation between classes. The output dimension is usually at most one less than the number of possible classes and makes the most sense for multiclass problems such as the student dataset. This could be why that dataset saw a larger improvement in clustering when compared to the wine dataset. Since the wine dataset has only binary classes, the LDA algorithm may have been too strong for the dataset.

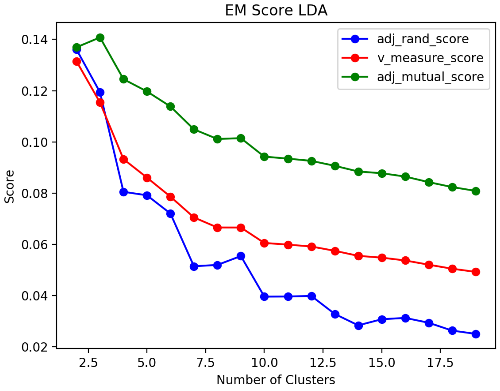
Student

Wine



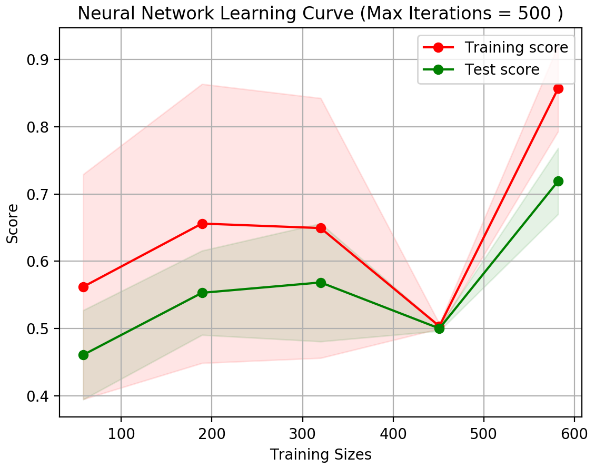






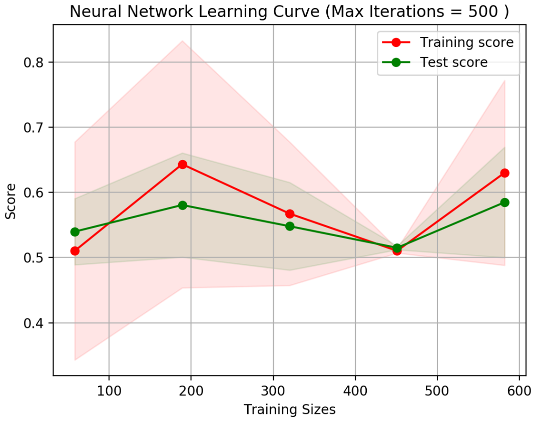
Neural Network Experiments

After applying dimensionality reduction to the student dataset, I wanted to see how a neural network would perform on this reduced dataset. I also wanted to see how adding the cluster memberships to the reduced dataset that were given from K-Means and EM affected neural network performance. For comparison purposes, I ran the neural network on the dataset without applying dimensionality reduction and without adding the cluster memberships to the dataset. The learning curve for this run is shown to the left.

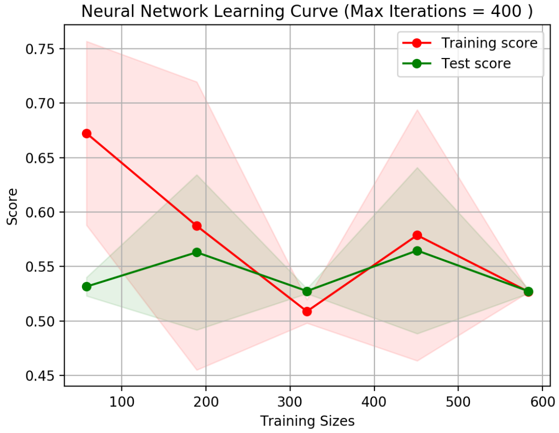


PCA

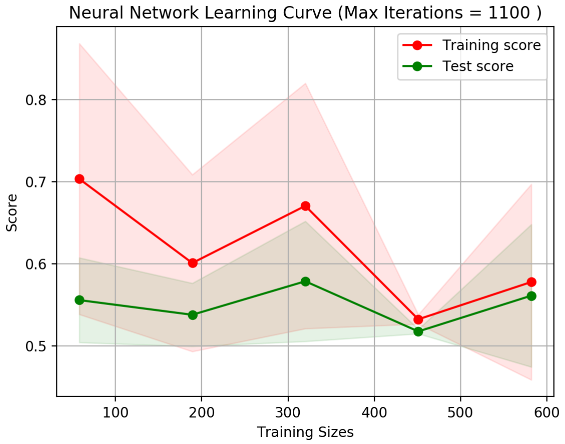
PCA Reduced Data



PCA + K-Means Clusters



PCA + EM Clusters

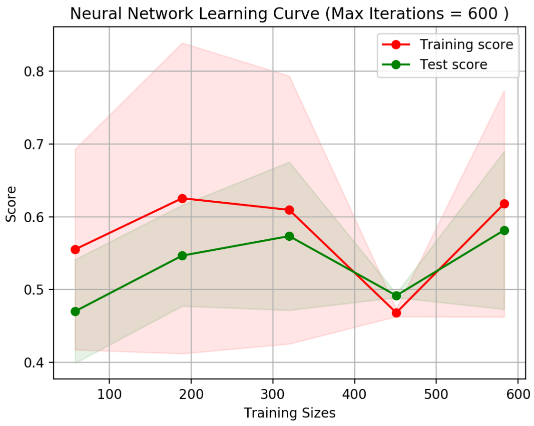


We see that the neural network performed slightly worse on the PCA reduced dataset when compared to the original dataset as training and testing scores converge on a lower accuracy rate. There was however less variance between the training and testing scores which indicates that the neural network converged faster on the reduced dataset. This result makes sense as PCA will create linear combinations of redundant features and gives features that maximize variance which would allow the neural network learner to converge at a faster rate. Adding the clusters found from K-means and EM made the neural networks perform considerably worse than the unmodified dataset.

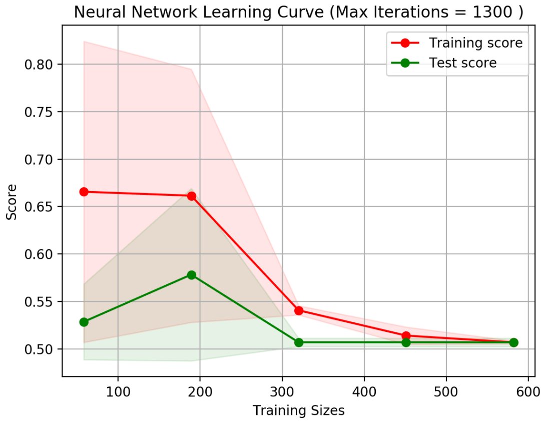
While knowing the clusters of the dataset gives insight on how the data is separated, adding it to this dataset may add more noise for the neural network learner thus making it harder to accurately predict the class of a given sample.

ICA

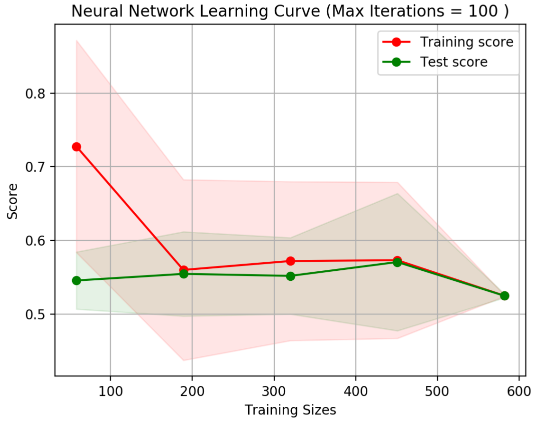
ICA Reduced Data



ICA + K-Means Clusters



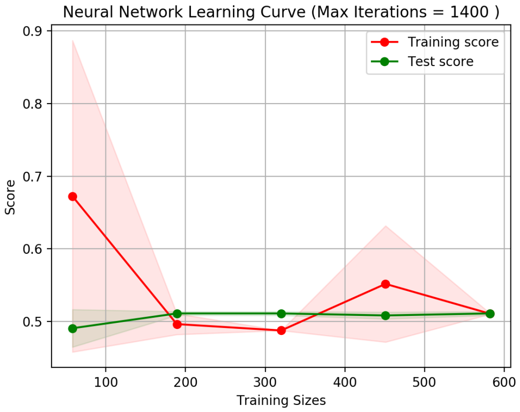
ICA + EM Clusters



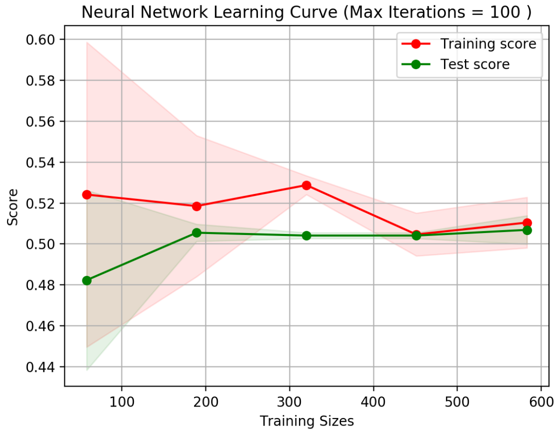
Using ICA alone on the dataset resulted in performance that was slightly worse than PCA. Since ICA tries to separate the dataset into features that are maximally independent from each other, a lot of features that may be more useful for the neural network will be lost after projecting the data into its independent parts. We also notice that adding the clusters found using K-means and EM to the dataset negatively impacted the performance of the neural network on the dataset as the training and testing scores both converged to lower values than the unmodified dataset which indicates under fitting. While using ICA is useful in finding more distinct and homogenous clusters, it does not seem to help a neural network classifier perform better and instead hinders its performance.

Random Projections

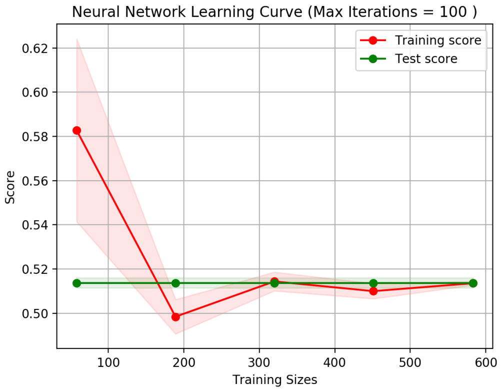
Random Projection Only



Random Projection + K-Means



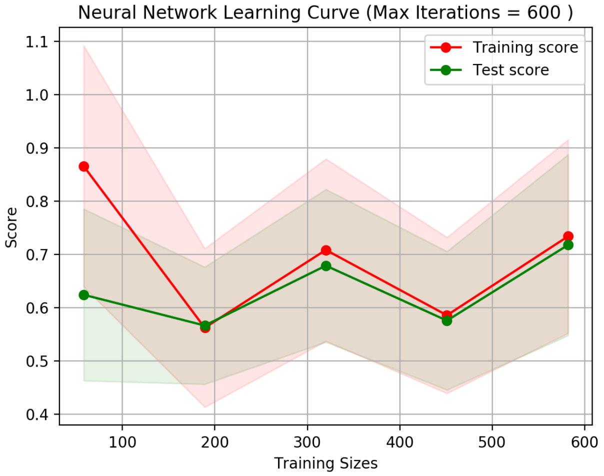
Random Projection + EM



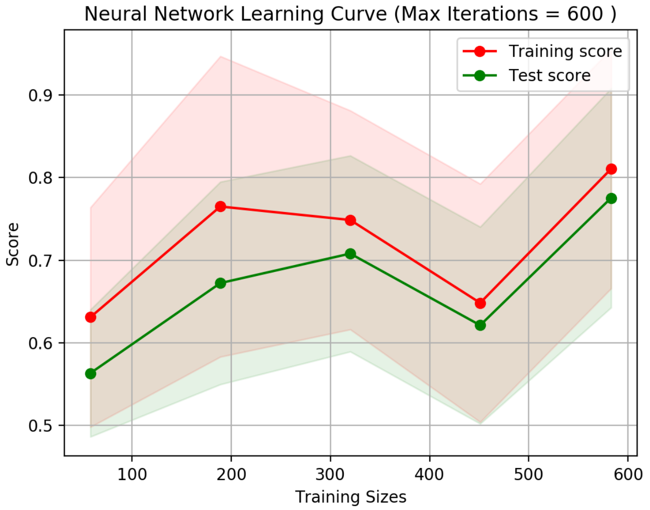
Using random projections on the student dataset resulted in the worst performance across the four dimensionality reduction algorithms used. As described earlier in the clustering experiments, random projections could be combining features that are not well correlated with each other and is possibly losing a lot of information that is relevant to the neural network. Thus, it is creating features that do not give useful information to the neural network and is also taking away features that contain a considerable amount of useful information which leads to the poor performance of the network. Also, since random projections negatively impacted the performance of the clustering algorithms used earlier, it is not much of a surprise that adding these clusters to the feature set negatively impacted training and testing scores of the neural network.

Linear Discriminant Analysis

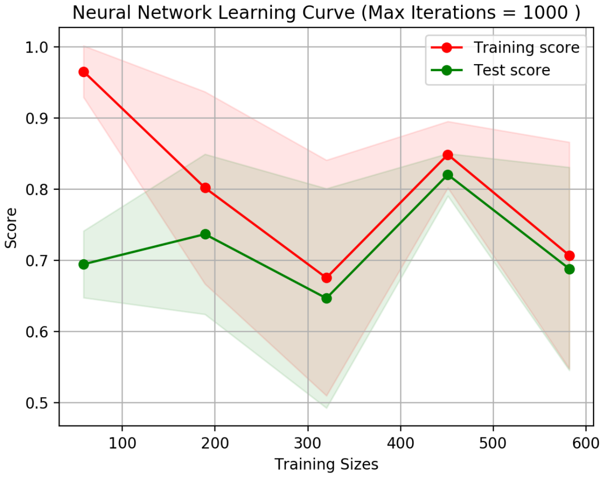
LDA Only



LDA + K-Means



LDA + EM



Linear Discriminant Analysis resulted in neural network performance that was slightly better than the performance on the unmodified dataset. When we only use LDA on the dataset, we see that the training and testing scores have very little variance as more data is given to the neural network and that they converge to scores that are slightly higher than the unmodified dataset. We can also see that adding the K-Means clusters to the dataset helped improve performance even more but did increase the variance slightly. Adding the EM clusters seemed to help performance somewhat but resulted in decreasing training and testing scores as more data was given to the neural network. The improvement in performance can be explained by the fact that LDA considers the labels of the dataset when it is reducing dimensions. This property results in the smallest feature set that maximizes separation between the different labels of the dataset, ultimately giving the neural network or possibly any other classifier enough data it needs to accurately predict the classes of samples in the testing set.