Stanley “Mason” Buchanan
CS 4641 Assignment 3
Analysis

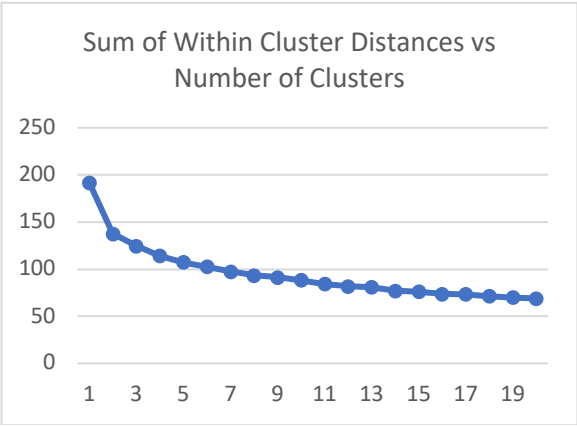
Unsupervised Learning and Dimensionality Reduction

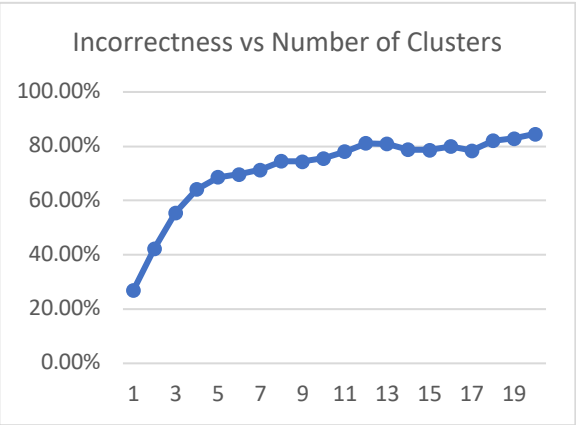
Introduction: The first dataset that was used in this project was one based on the performance of Portuguese students in math classes. The second dataset that was used was one based on travel review data from people visiting East Asian countries. As stated in my previous analysis, the Student dataset was comprised of 649 instances with 33 attributes prior to the introduction of one-hot encoding, while the Travel dataset had 980 instances spread across 11 attributes. These are interesting primarily because they represent data that would help people to make everyday decisions; one could determine at-risk students based on the various characteristics associated with them, and one could use travel reviews to determine whether or not they would like to visit a location. The output feature for the student dataset was whether or not the student had access to internet at home and the output feature for the travel dataset was whether the religious institutions of the region were rated positively or not.

Clustering:

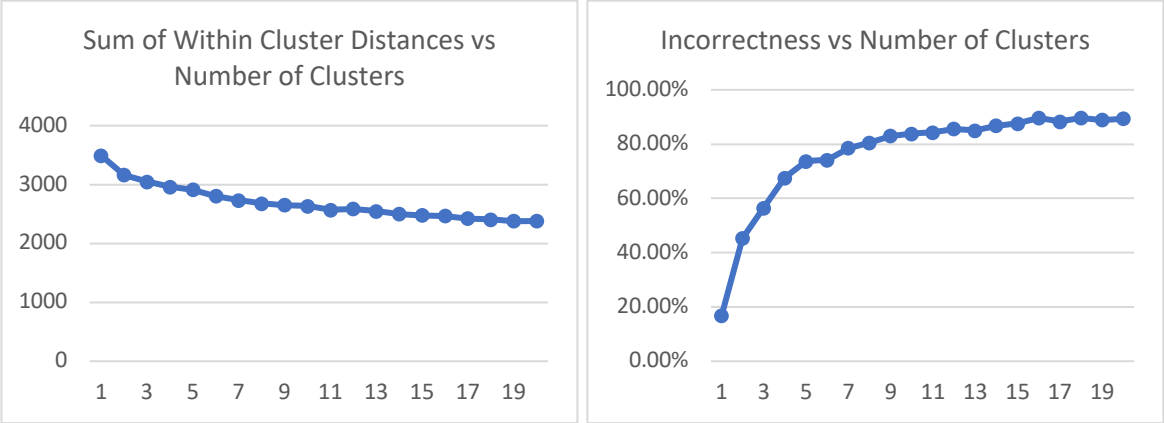
K-Means: While running k-means, I used the default hyperparameters used by WEKA, only varying the number of clusters. I also decided to use Euclidean distance as my similarity metric instead of Manhattan distance due to consistently better clustering results. To determine the optimal number of clusters, I utliized the elbow method as a quick, qualitative technique that provides solid results approximating more complicated, mathematically heavy methods. For both of the datasets, k = 4 turned out to be a reasonable value.

Student Dataset:





Travel Dataset:



For k-means, the primary measure of performance was the sum of within cluster distances. Ideally, this number will be reasonably small, indicating a high degree of similarity between adjacent points in a cluster. I also plotted how correctly the algorithm clustered the data relative to the output for each dataset. This is not really a measure of performance, so much as it helps in relating the topic of clustering to previously discussed topics such as classification.

The algorithm performed as expected, with within cluster distances steadily decreasing as the number of clusters increases. This makes sense due to the fact that as you increase the number of clusters to sort the data into, the smaller those clusters are likely to be. Many of the clusters actually had no data assigned to them, indicating that there are only a few important features that provide the majority of the information in the dataset.

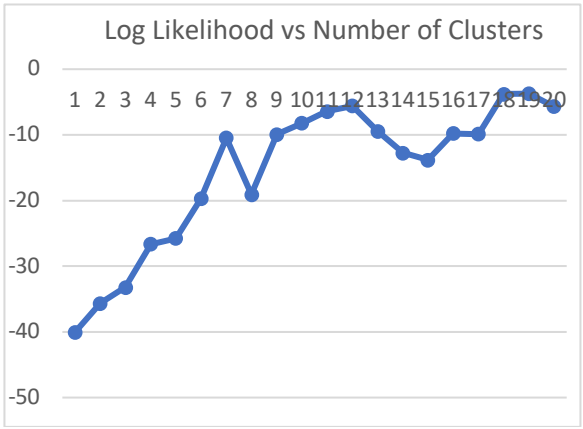
Interestingly, when the graphs for the clusters were produced, it was very unclear as to what the centroid of the clusters were. This is likely due to relatively low correlations between features. This is shown in the correlation matrix for the travel dataset. The student dataset had too many features to include the matrix in this document, but it can be found in the accompanying excel spreadsheet that documents my results. As can be seen below, only two relationships has a correlation greater than 0.5, indicating a low level of correlation between features in the dataset. Alongside this, the clusters did not line up with the labels of the datasets, indicating relationships that were not previously noticed.

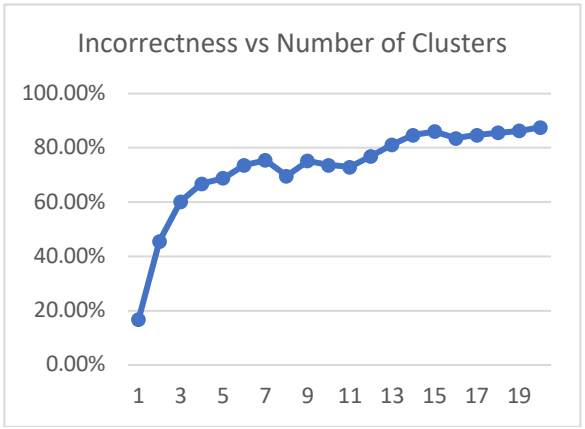
Correlation matrix

1	-0.19	0.01	0.07	-0.1	0.09	-0.01	0.02	-0.05
-0.19	1	0.04	0.13	0.12	0.15	0.11	-0.16	0.07
0.01	0.04	1	0.06	0.28	0.36	0.75	-0.17	-0.09
0.07	0.13	0.06	1	0.1	0.22	0.23	-0.1	0.03
-0.1	0.12	0.28	0.1	1	0.58	0.23	-0.02	0.04
0.09	0.15	0.36	0.22	0.58	1	0.43	0	0.1
-0.01	0.11	0.75	0.23	0.23	0.43	1	-0.07	0.08
0.02	-0.16	-0.17	-0.1	-0.02	0	-0.07	1	0.17
-0.05	0.07	-0.09	0.03	0.04	0.1	0.08	0.17	1

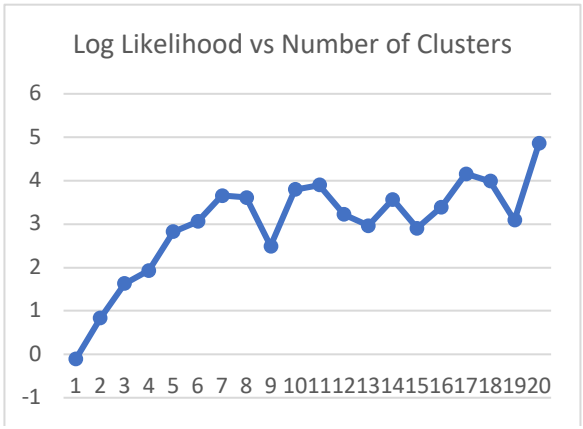
Expectation Maximization: Similar to k-means, I utilized WEKA's default hyperparameters, only varying the number of clusters, and I used the elbow method again to determine a near-optimal value for the number of clusters. For the student dataset, k = 6 was a good value, and for the travel dataset, k = 8 turned out to be a good value.

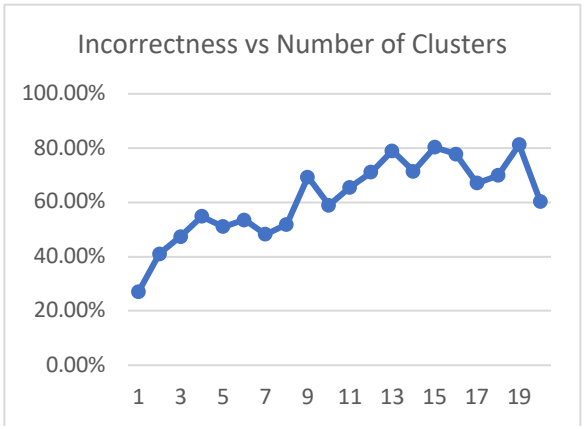
Student Dataset:





Travel Dataset:





For expectation maximization, the primary measure of performance was log likelihood. This number is to be maximized to reflect an increase in how well the clustering algorithm fits the dispersion of the data. Again, I measured the incorrectness of clustering with respect to the output of the dataset to provide a nice reference point to my previous analysis.

The clusters produced by expectation maximization closely mirrored those made by k-means, so there is not much more to be said about the properties of the clusters. Again, the correlations between features showed up by making it very unclear as to what the centroid for each cluster was. However, what was clear was that the clusters did not necessarily line up with the labels, which showed that there may be some interesting relationships that would arise during feature transformation.

Dimensionality Reduction:

All of the dimensionality reduction algorithms helped to decrease the overall number of features present in the datasets, aiding with the issue of the curse of dimensionality. Here I will provide a brief overview of the different dimensionality reduction algorithms used in this project before diving into the performance variations they provided during clustering.

Principal Components: Principal Components analysis (PCA) performed as expected, generating features written as linear combinations of the previous features. The correlation matrix for the new features was the identity matrix, reflecting how the transformation of the original features worked to create only the features that are most important to providing predictive power.

Independent Components: Independent Components analysis (ICA) performed as expected, generating features that had high levels of independence from one another. This is indicated in the fact that the correlation matrix for the new features was the identity matrix.

Random Projection: Random Projection (RP) worked as expected, taking random features from the dataset and creating a subset. It did not transform the features in any way, and, since it takes a subset, this is what leads to the decrease in the number of features. As there was no feature transformation, the correlation values between the selected features did not change. Due to the low values of the correlation matrix for each dataset, running random projection multiple times did not have significant effects on the level of independence between features. In very few cases, there were slight improvements, but I settled with a more average case of RP to show its differences from the next algorithm used, Correlation-Based Feature Selection (CFS).

Correlation-Based Feature Selection: Correlation-Based Feature Selection (CFS) was a built-in feature selection method included in WEKA that looked to create the subset of features that is optimal based on their correlations with one another, but without transforming the original features. Because there was no feature transformation, the correlation values between the selected features did not change.

Clustering w/ Dimensionality Reduction:

Dataset	Clustering Algorithm	Dim. Reduction Algorithm	Variance	Incorrect %	Pre-Dim. Reduction Variance	Pre-Dim. Reduction Incorrect %
Travel	EM	PCA	-10.14734	61.94%	3.61475	51.94%
Student	EM	PCA	-45.58802	63.54%	-19.69505	73.67%
Student	K-Means	PCA	327.764092	67.09%	2961.579254	67.59%
Travel	K-Means	PCA	99.9044521	58.98%	114.0578749	64.18%
Travel	EM	ICA	19.14832	53.47%	3.61475	51.94%
Student	EM	ICA	66.39712	56.20%	-19.69505	73.67%
Student	K-Means	ICA	428.378349	56.71%	2961.579254	67.59%
Travel	K-Means	ICA	137.800881	54.39%	114.0578749	64.18%
Travel	EM	RP	-10.25163	76.94%	3.61475	51.94%
Student	EM	RP	-31.71571	71.39%	-19.69505	73.67%
Student	K-Means	RP	52.6319938	68.10%	2961.579254	67.59%
Travel	K-Means	RP	104.921585	64.69%	114.0578749	64.18%
Travel	EM	Greedy CFS	4.44226	52.45%	3.61475	51.94%
Student	EM	Greedy CFS	6.36108	55.19%	-19.69505	73.67%
Student	K-Means	Greedy CFS	93.1818411	60.25%	2961.579254	67.59%
Travel	K-Means	Greedy CFS	35.3334697	63.57%	114.0578749	64.18%

PCA: For PCA, the clusters were different than before, primarily because through it, the dataset underwent feature transformation. Interestingly enough, PCA did not perform well with EM, and actually caused a decrease in log likelihood. However, for k-means, it generated a decrease in the sum of within cluster distances, but the student dataset showed much more change compared to the travel dataset. This is likely due to the number of features being higher for the student dataset, so the effect of decreasing the number of features will be felt much more acutely.

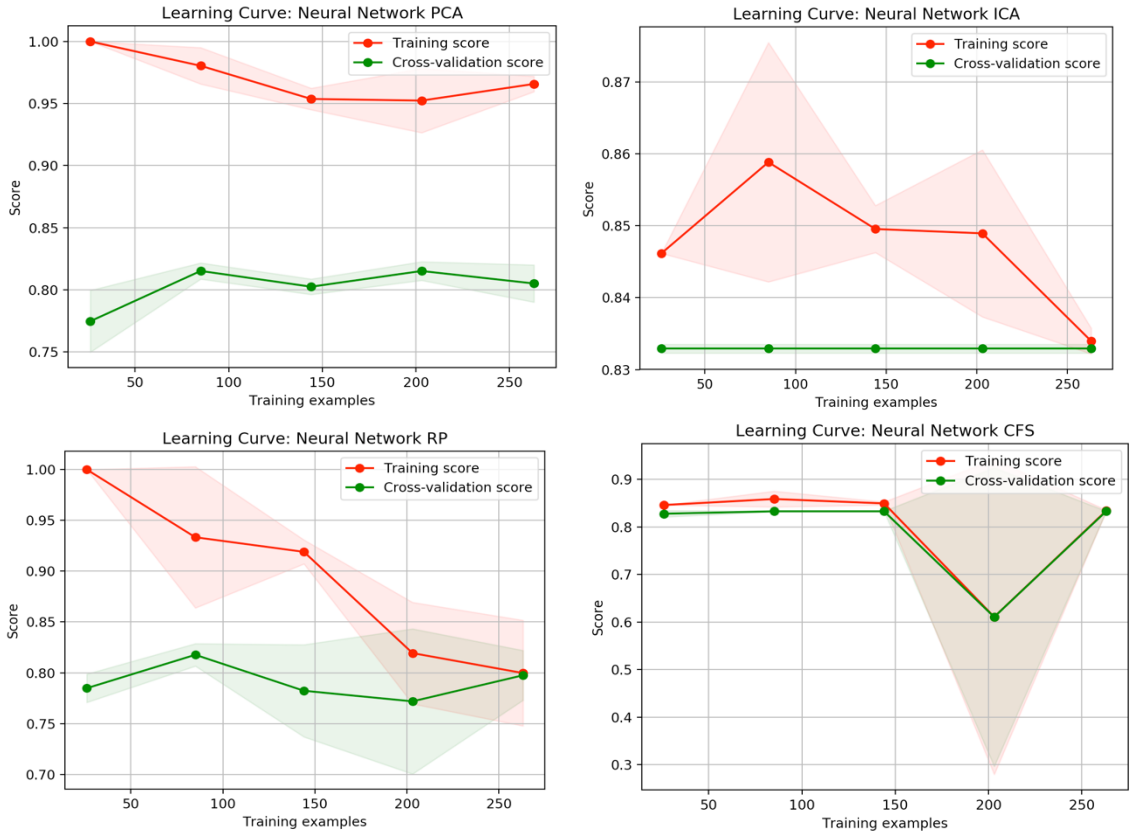
ICA: Similar to PCA, ICA produced clusters that were different, due to the feature transformation that occurred. It worked well in all cases except for the case of k-means with the travel dataset, which is an interesting one in that increased in intra-cluster variance but decreased in the percent of instances that were incorrectly placed based on the output variable. It is curious to me that PCA and ICA worked well on opposite things, given that they both built independent feature sets, which brings to mind an important idea: that there are multiple viable independent feature sets that will vary in performance depending on the algorithm a.k.a. no free lunch.

RP: RP did not perform very well with EM, likely due to a subset that decreased predictability in the features' correlation matrix. Similar to PCA, it did well with k-means, but primarily impacted the results of the student dataset due to an alleviation of the curse of dimensionality. Unlike PCA, RP produced clusters with high inter-cluster distances, due to the fact that it did not alter the correlation matrix for each dataset to the point of being the identity matrix.

CFS: CFS worked very well with both datasets and generated clusters that had decreased variance for both EM and k-means. It did not perform as well as ICA for EM, but still outperformed both RP and PCA in that regard. Only RP performed better with k-means for the student dataset, and in all other cases of k-means, CFS was the clear winner in terms of intra-cluster variance. It is interesting to me that it performed better all-around than PCA and ICA

given that these two algorithms actively seek to build the best features they can; perhaps some nuance is lost in the feature transformation that CFS is still able to access.

Neural Network w/ Dimensionality Reduction:



Dataset	Dim. Reduction Algorithm	Test Set Accuracy	Time (s)	Original Test Set Accuracy	Original Time (s)
Travel	PCA	86.73%	1.409734964	86.86%	1.0491
Travel	ICA	87.04%	1.324681044	86.86%	1.0491
Travel	RP	84.57%	1.112987041	86.86%	1.0491
Travel	Greedy CFS	86.73%	0.978299856	86.86%	1.0491
Student	PCA	84.73%	2.823863983	81.94%	1.821
Student	ICA	81.68%	2.217214823	81.94%	1.821
Student	RP	83.97%	0.283557177	81.94%	1.821
Student	Greedy CFS	83.97%	0.354314804	81.94%	1.821

After running the neural network on both datasets, I decided to make the student dataset the focus of my analysis as it underwent more dramatic changes due to the dimensionality reduction.

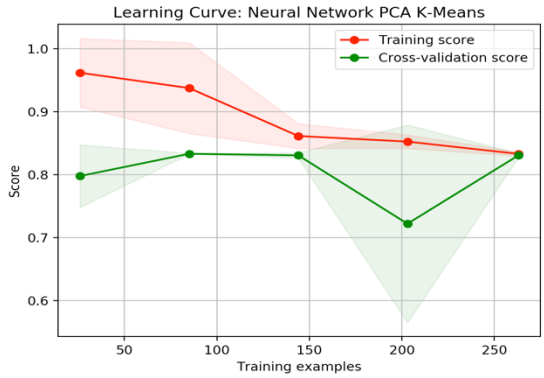
PCA: PCA caused an increase in accuracy, but also an increase in time. It seems that the introduction of these new features, though independent, only increased the difficulty of creating a viable model for the neural network. It performed in a fairly standard way graphically speaking, as the number of training examples increased, so too did the level of overfitting, as seen by the divergence between the two lines at the end of the graph.

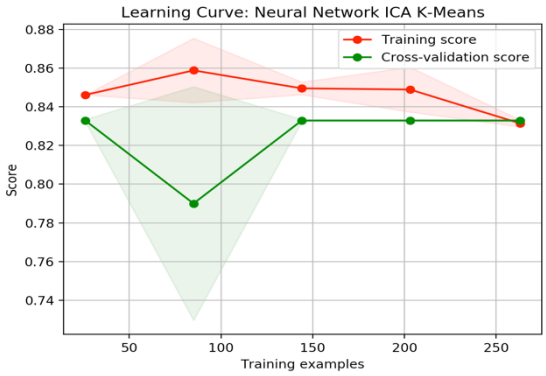
ICA: Interestingly, ICA did worse than the original features. It seems to have run into the same problem as PCA, but what is odd is that the features did worse even though they have, theoretically speaking, been transformed into a more "optimal" form. There may be some hidden nuances in the original features that the neural network could pick up on that were eliminated in the feature transformation. Regarding overfitting, ICA had some interesting results; it appears that the testing set contained more of the hard examples, whereas, early on, the testing set was made up of easier examples. This is typically something that can lead to the behavior seen in the graph where the training set performance decreases as the number of examples increases.

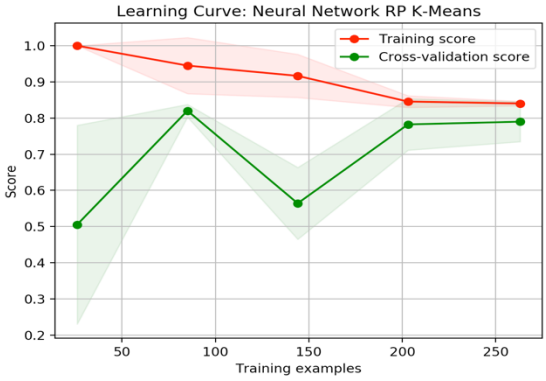
RP: RP showed an improvement in both accuracy and time. This really goes to show the importance of the curse of dimensionality, and although RP could be considered rather "dumb," the fact that it outperforms PCA and ICA, which could be considered "smarter," depicts more vividly the no-free-lunch theorem. Feature transformation may have the effect of sterilizing the data to the point where it may lose some value in terms of providing predictive power. RP had similar overfitting characteristics to ICA, as both seem to have easier examples given to the training set early on, while the testing set is given more difficult examples.

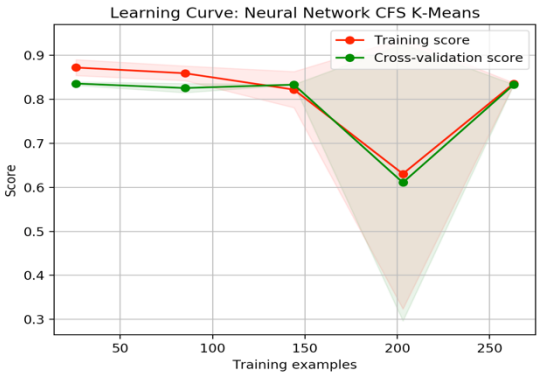
CFS: Similar to RP, CFS produced improvements in both accuracy and time. Again, this shows the importance of the curse of dimensionality, as a simple ranking of the most correlated features is able to reduce a lot of redundancies without necessarily compromising too much and removing some of the finer details necessary to develop a comprehensive model, such as what was seen with PCA and ICA in this case. CFS had a very interesting graph that goes to show how powerful it can be to reduce to key features. The two lines follow each other almost exactly, indicating a very strong fit and high level of generality. This is very encouraging and is likely explained by the lessening of the effects of the curse of dimensionality.

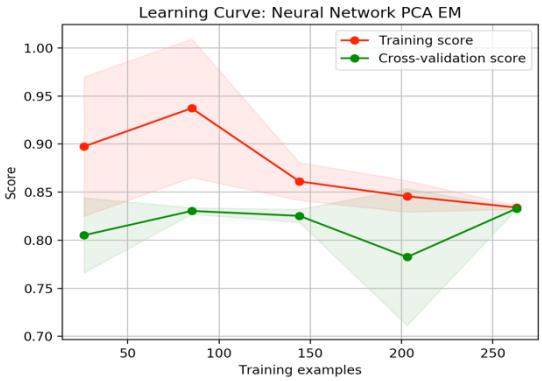
Neural Network w/ Dimensionality Reduction and Clustering:

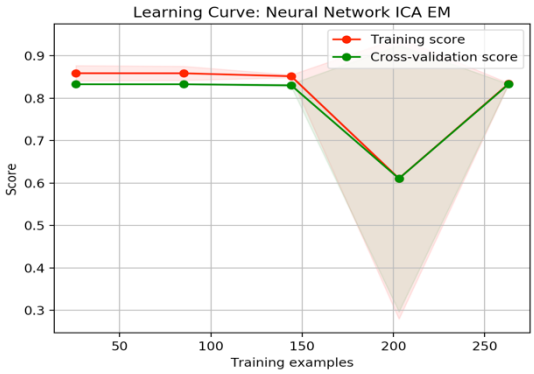


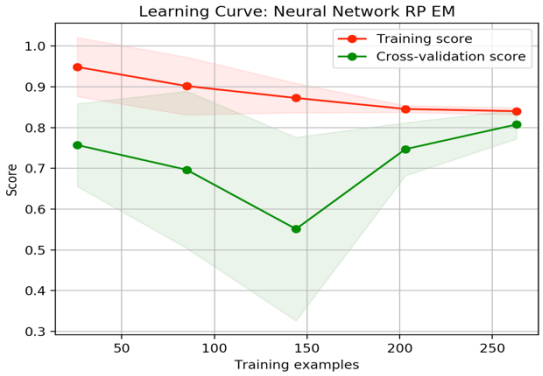


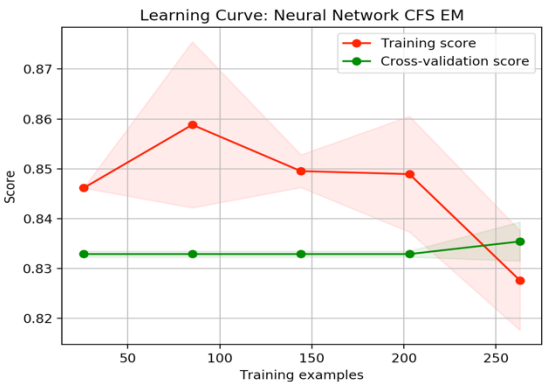












Dataset	Dim. Reduction Algorithm	Clustering Algorithm	Test Set Accuracy	Time (s)	Pre-Clustered Test Set Accuracy	Pre-Clustered Time (s)	Original Test Set Accuracy	Original Time (s)
Student	PCA	EM	80.92%	2.36458111	84.73%	2.823863983	81.94%	1.821
Student	PCA	K-Means	80.15%	2.45732594	84.73%	2.823863983	81.94%	1.821
Student	ICA	EM	80.15%	2.29760504	81.68%	2.217214823	81.94%	1.821
Student	ICA	K-Means	79.39%	2.55164504	81.68%	2.217214823	81.94%	1.821
Student	RP	EM	83.97%	0.30196214	83.97%	0.283557177	81.94%	1.821
Student	RP	K-Means	83.97%	0.27864003	83.97%	0.283557177	81.94%	1.821
Student	Greedy CFS	EM	83.97%	0.33182383	83.97%	0.354314804	81.94%	1.821
Student	Greedy CFS	K-Means	83.97%	0.35121894	83.97%	0.354314804	81.94%	1.821

PCA: For EM and PCA as well as k-means and PCA, the neural network performed more poorly both in terms of accuracy as well time. This is likely due to the sterilization of the feature set that I discussed in previous sections. What is interesting to me is that the inclusion of the clusters only decreased the performance of the neural network. However, upon further analysis, it begins to make sense why that is the case. EM and PCA performed very poorly in clustering with respect to the output variable. Thus, the inclusion of the clusters as a feature in the dataset likely created "confusion" in the neural network in the sense that the clusters presented irrelevant information that only provided confounding information to the network. This shows that it is likely that the clusters produced by EM and k-means when using data manipulated using PCA captured little of the relationship between students and their access to the internet at home. This thought is bolstered by the fact that the neural network performed much better on the pre-clustered dataset in terms of accuracy. Both PCA with EM and PCA with k-means exhibited similar behavior regarding overfitting. They both had convergence between the training and testing lines, indicating that the testing set had easier examples to start and that the testing set had more difficult ones.

ICA: Similar to PCA, ICA with EM and ICA with k-means performed rather poorly when utilized on the dataset and then being used in the neural network. Both in terms of accuracy and time, these setups performed less well than the network being run on the pre-clustered dataset. I believe that this is because of the same reasons outlined regarding PCA. It seems that on datasets like the student dataset, feature transformation may be an unnecessary overcomplication that introduces confounds to the experiment that the network may not be able to account for. ICA with k-means had the familiar convergence pattern we have been seeing, indicating that the testing set had easier examples to start and that the testing set had more difficult ones. ICA with EM exhibited behavior indicating a very good fit and high level of generality as the lines match each other very closely from the start.

RP: RP is very interesting in this case, as the network performed just as well on the clustered dataset as it did on the pre-clustered dataset with regards to accuracy. I think that this further supports the ideas I presented in the previous section relating to the tangible effects of dimensionality reduction on smaller datasets in helping to alleviate the curse of dimensionality. The fact that the introduction of the clusters as a feature had little to no effect even when they were very inaccurate with respect to the output variable shows that, even with the introduction of this confound, the shrinking of the feature set provided a much greater effect, even if the subset of features was selected arbitrarily. The fact that it performed much faster also indicates that using RP to get rid of redundant features helped in reducing the model complexity necessary to accurately predict the output variable. RP with k-means and RP with EM also exhibit the convergent behavior we have been seeing, indicating that the testing set had easier

examples to start and that the testing set had more difficult ones.

CFS: CFS performed very similarly to RP and also had no change in accuracy with the introduction of the clusters to the dataset. However, it did have a decrease in time, which indicates that CFS combined with EM and k-means produces clusters that relate very well to the output variable. This is actually seen in the section detailing the clustering algorithms after feature selection. CFS with k-means fit well, with both lines closely mirroring one another, indicating a strong fit and high level of generality. CFS with EM was interesting and seemed to be a stronger version of the convergent behavior seen in previous sections. This is potentially due to the fact that each feature is much more important after using CFS, since it reduced to a smaller subset of features, and that caused a more extreme version of the convergent behavior we have been seeing.