Unsupervised Learning & Dimensionality Reduction

The Datasets

The two datasets chosen are 1.) red wine quality and 2.) handwritten digits. For the wine dataset, 11 different features are given, and the goal is to identify whether wine is of good or bad quality making it a binary classification problem. With 11 features present and 1599 samples, there is room to possibly apply dimensionality reduction and see if the model can be created in a way that reduces overfitting by using less features, or if the time taken to train a supervised learner can be reduced.

The handwritten digits database uses 64 different features and the goal is to correctly label which digit from 0-9 has been written. This problem is interesting for unsupervised learning because of the large number of features present from which we can apply dimensionality reduction.

Both datasets are well balanced as can be seen in the plots below showing the distributions of the labels.

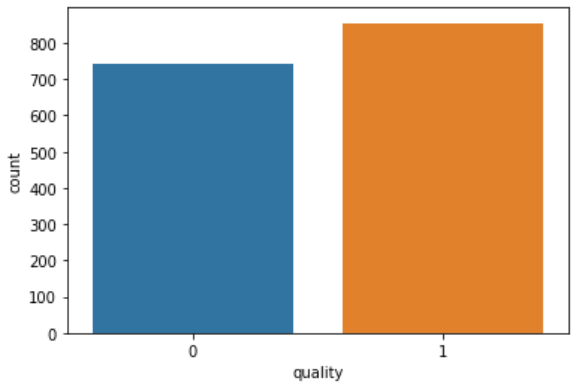


Figure 1: Distribution of wine quality, 0 being bad and 1 good

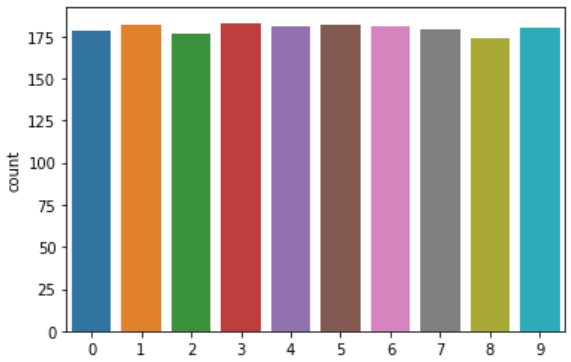


Figure 2: Distribution of handwritten numbers in digits dataset

Clustering

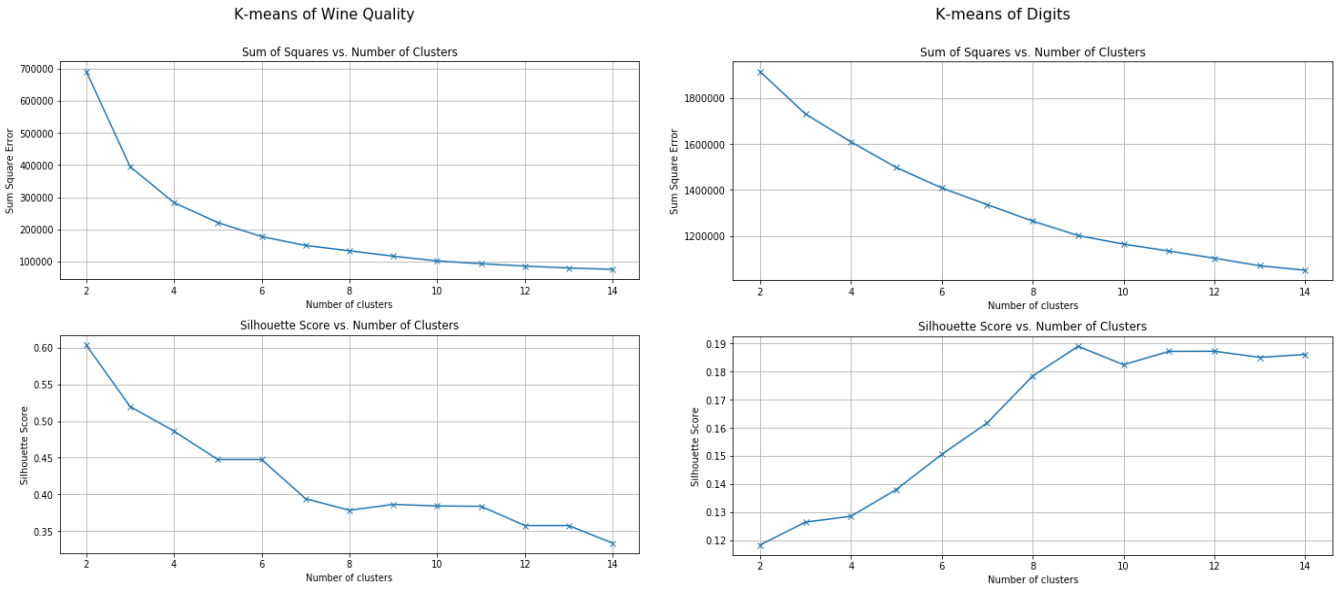
For this assignment, we are using 2 feature transformation algorithms, k-means and expectation maximization, in order to learn from the data to glean what the important information may be based on the distributions of the data. This is different from our previous work where we were defaulting to using all of the available data and assuming that the learner would either choose to use or not use the available features based on things like the information gain the attribute would provide.

K-means

The first clustering algorithm that was run is the K-means algorithm, which tries to sort the data into *k* different clusters. *K* was chosen by looking at both the inertia of the model and the silhouette score and by using the Elbow Method. The inertia of the datapoints is equal to the mean squared distance between each sample and the centroid of the cluster it belongs to, shown on the following plots as the sum square error. The lower the average inertia is in the model would imply that the clusters created are tightly formed, which is desired. The silhouette score on the other hand puts a value on the separation between different clusters. A low silhouette score would indicate that the number of clusters

chosen is not optimal. The data for the features of each dataset was scaled such that the Euclidean distance between points could be used as a metric for distance.

For interpreting the values, the “Elbow Method” was used. This method involves looking at the curve of the graph and trying to determine where the “elbow” is. This means that we are trying to pick an optimal point at which adding more clusters only provides marginal value. For the case of the sum of the square errors showing the inertia of the k-means data, it is the point at which the inertia is decreasing but no longer at a meaningful rate.

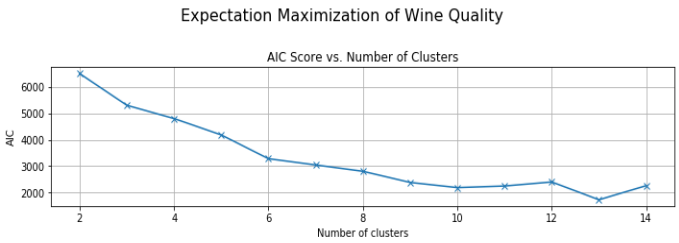


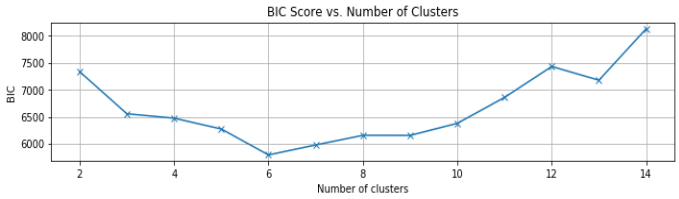
For the wine quality dataset, it was determined that 2 or 3 clusters would be best based on the elbow in the inertia chart combined with the higher silhouette score. For the digits database, the elbow in the inertia plot is around 9 or 10, which is where the silhouette score starts to level off.

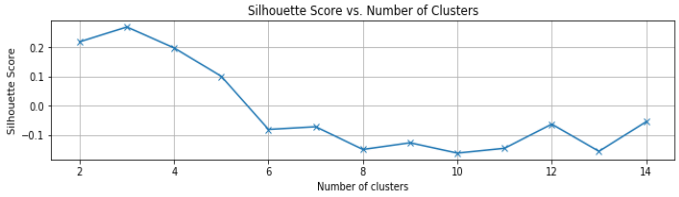
Expectation Maximization

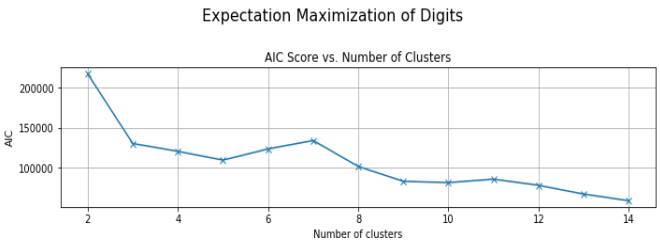
The second method that was used for clustering was the Expectation Maximization method. The values of merit in this case were the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), and again the silhouette score. The AIC score is used to represent the amount of information lost by a given model, and thus should be minimized. The BIC score is used to determine when adding more parameters to a model may result in overfitting and should also be minimized. The silhouette score was described in the previous section.

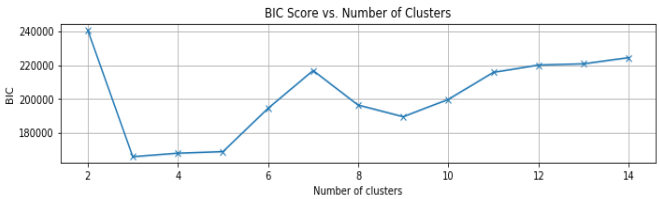
For the wine quality dataset, in trying to balance out having a lower AIC and BIC score along with a higher silhouette score, the optimal number of clusters looks to be between 2-4, as seen by the plots below. For the digits database, there is a local minimum of the AIC and BIC scores around 10 which coincides with the highest silhouette score.

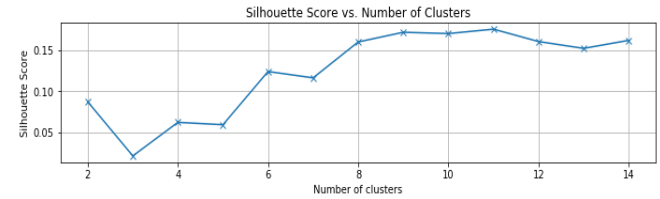












Dimensionality Reduction

Using dimensionality reduction on the two datasets, we can try to protect ourselves from the curse of dimensionality in which as we add more features, the amount of data that we need grows exponentially. By trimming down the features to just the useful ones, we can reduce the amount of data needed and hopefully make the learning problem easier.

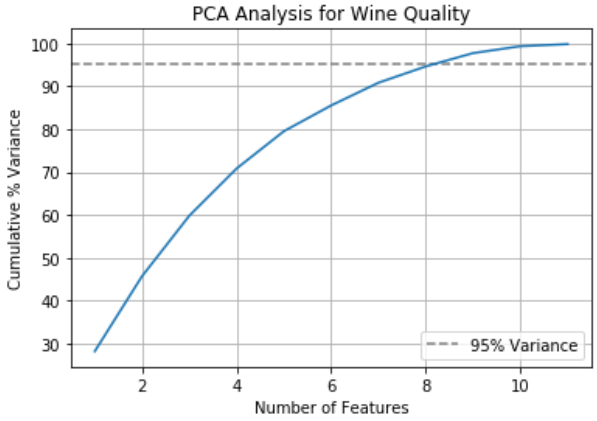
For this assignment, four algorithms were used: Principal Components Analysis (PCA), Independent Components Analysis (ICA), Randomized Components Analysis (RP), and Univariate Feature Selection (UFS).

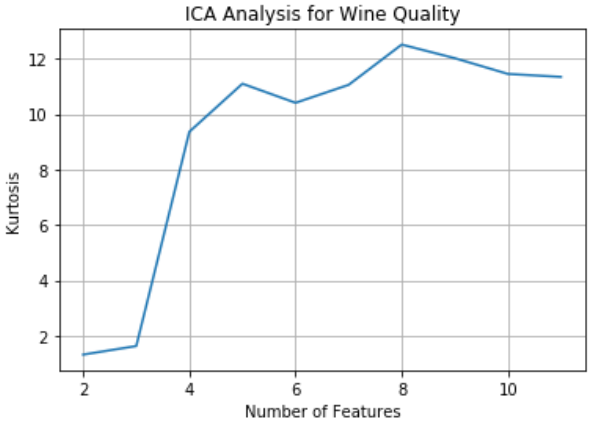
For PCA, the cumulative sum of variance (eigenvalues) was used in order to determine the best number of features. A threshold was set at 95% variance of the dataset, after which it was determined that adding more features would generate marginal return.

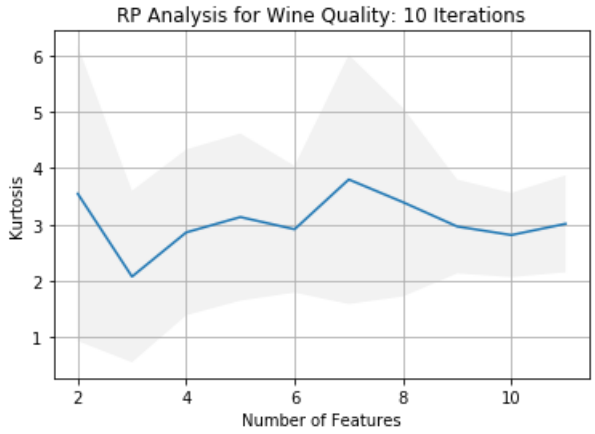
For ICA, the kurtosis of the distributions was calculated and the number of features that generated the largest kurtosis was chosen. In maximizing the kurtosis, we are choosing a distribution that is very pointy, or a supergaussian. The same approach was also taken for RP.

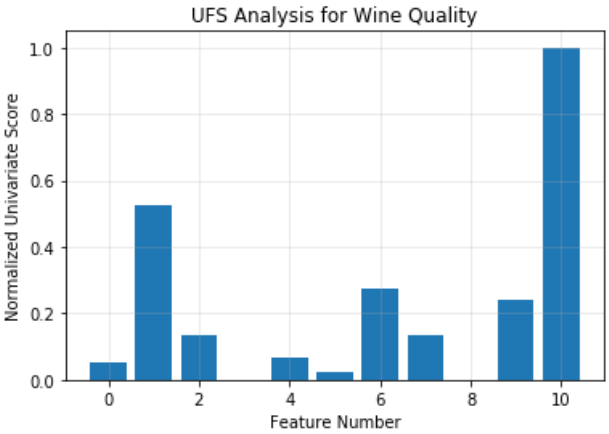
For UFS, the normalized univariate score was taken and then a more qualitative approach was taken to determine which features were of value.

Wine Quality





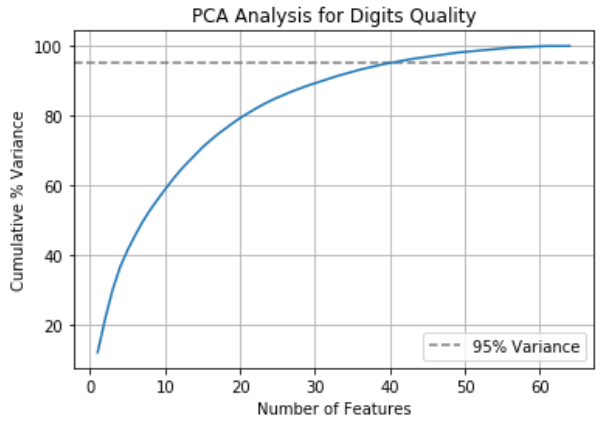


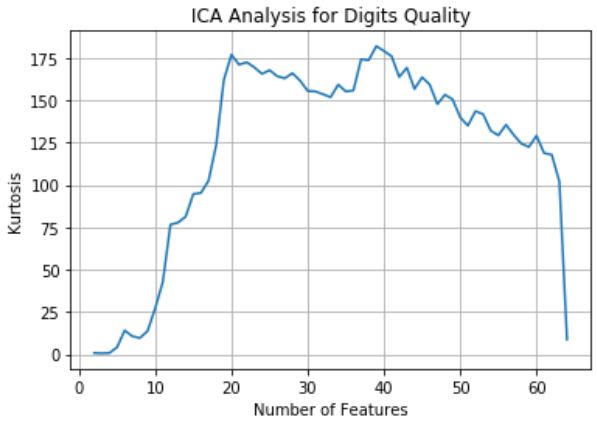


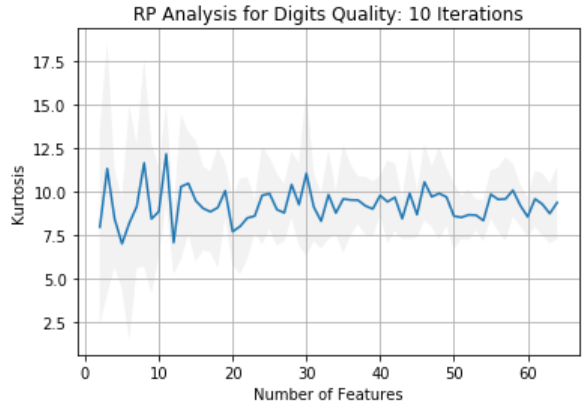
	PCA	ICA	RP	UFS
Number of Features	9	8	7	4
Total Time (sec)	0.002	0.06	0.073	0.18
Reconstruction Error	5.3e-31	0.144	0.879	N/A

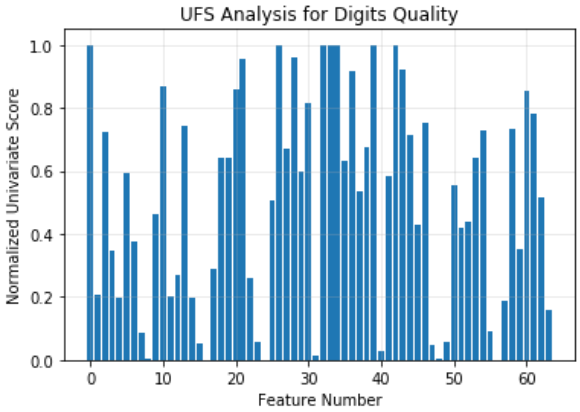
For PCA and ICA, the algorithms chose most all the features. PCA had the lowest amount of reconstruction error. With the ICA algorithm choosing 8/11 features, it doesn't appear that there is any real mutual information to take advantage of in this dataset. The RP algorithm was rather noisy and gave different answers each time with minimal amount of iterations. When more iterations were run, the kurtosis ended up becoming a flat line with no optimal number of features. This indicates that the features in the dataset may not be related to each other. The reconstruction error for RP was by far the worst. For UFS, most of the features play a minimal role in the result, so 4 features were chosen going forward using this method.

Digits









	PCA	ICA	RP	UFS
Number of Features	40	39	11	12
Total Time (sec)	0.008	11.121	0.882	0.238
Reconstruction Error	2.5e-30	0.059	0.803	N/A

PCA and ICA for the digits dataset again chose around the same set of features, which gives me confidence that they are probably picking the same features to use. In this case, they were both able to cut out about a third of the data needed. ICA took by far the longest to run. RP again had a very large reconstruction error and showed no clear value for the best number of features. When ran with more and more iterations, the kurtosis average out with a smaller standard deviation as more features were added. UFS in this case showed around 12 features having a large involvement in the result, so that value was used going forward with the rest of the project.

Clustering after Dimensionality Reduction

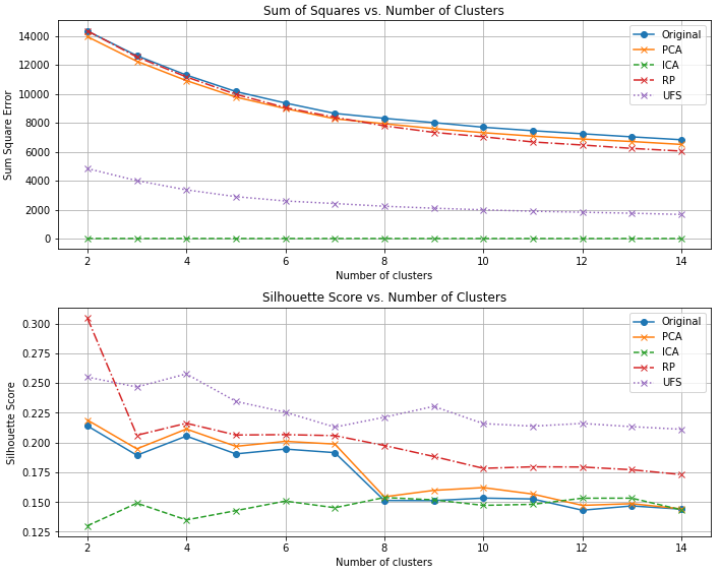
Wine Quality

For the k-means method, when using the reduced data to perform the clustering algorithm, the RP dimensionality reduction seemed to perform the best, both in time and silhouette score. As seen in the below charts, the ICA method resulted in the lowest inertia value, but suffered in terms of a silhouette score. The UFS method produced a better inertia with the same time as the RP method, but did not have as good of a silhouette score.

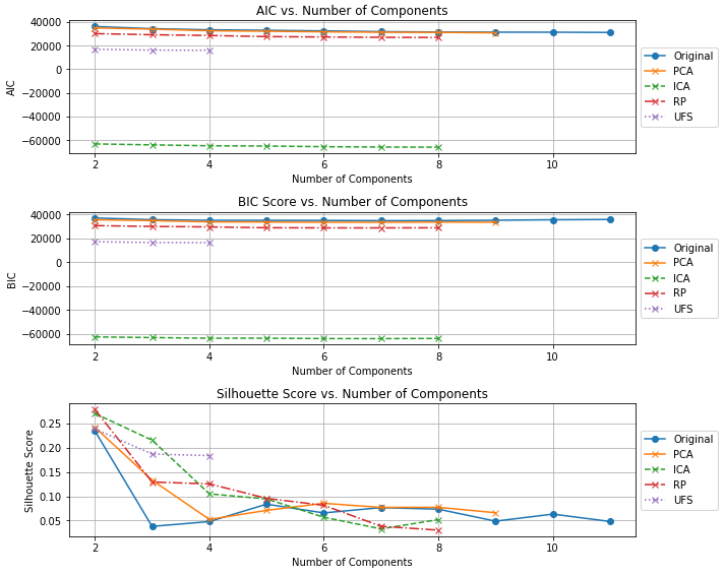
It seemed like all the methods followed the same general trend as far as inertia goes except the ICA and UFS method, which performed better.

```
--------------------------------------------------------------------------------
*  K-Means Wine Benchmark  *
--------------------------------------------------------------------------------
init           time    inertia homog   compl   v-meas  ARI     AMI     silhouette
--------------------------------------------------------------------------------
Original       0.05s   14330   0.025   0.025   0.025   0.027   0.025   0.207
PCA-based      0.04s   13945   0.029   0.031   0.030   0.030   0.030   0.219
ICA-based      0.06s   7       0.088   0.092   0.090   0.099   0.089   0.141
RP-based       0.03s   14349   0.002   0.002   0.002   0.000   0.002   0.305
UFS-based      0.03s   4835    0.147   0.149   0.148   0.178   0.147   0.255
```

K-means of Wine Quality Using Dimensionality Reduction



EM of Wine Quality Using Dimensionality Reduction




```
--------------------------------------------------------------------------------
*  EM Wine Benchmark  *
--------------------------------------------------------------------------------
init           time    aic     bic     homog   compl   v-meas  ARI     AMI     silhouette
--------------------------------------------------------------------------------
Original       0.17s   31421   34771   0.121   0.044   0.065   0.058   0.063   0.073
PCA-based      0.07s   31095   33456   0.132   0.049   0.072   0.065   0.070   0.077
ICA-based      0.11s   -65805  -63875  0.124   0.046   0.067   0.057   0.065   0.052
RP-based       0.08s   26806   28736   0.138   0.052   0.076   0.057   0.074   0.030
UFS-based      0.08s   15460   16100   0.200   0.071   0.105   0.076   0.104   0.104
```

For the expectation maximization method, UFS had the best silhouette score. The ICA method produced the lowest AIC and BIC scores though, indicating that this model would be the most generalized.

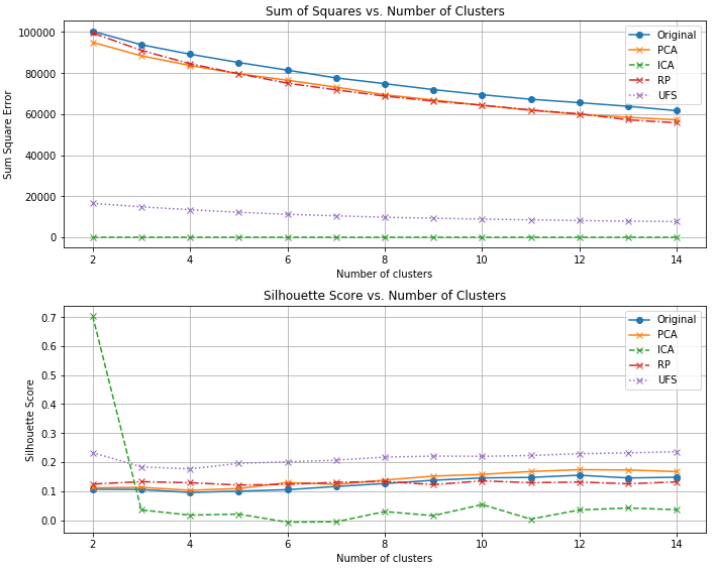
The takeaway for both methods seems to be that the ideal number of clusters would still be 2, which is the same as what was seen before any dimensionality reduction. This means that the same conclusion could be reached with less data.

Digits

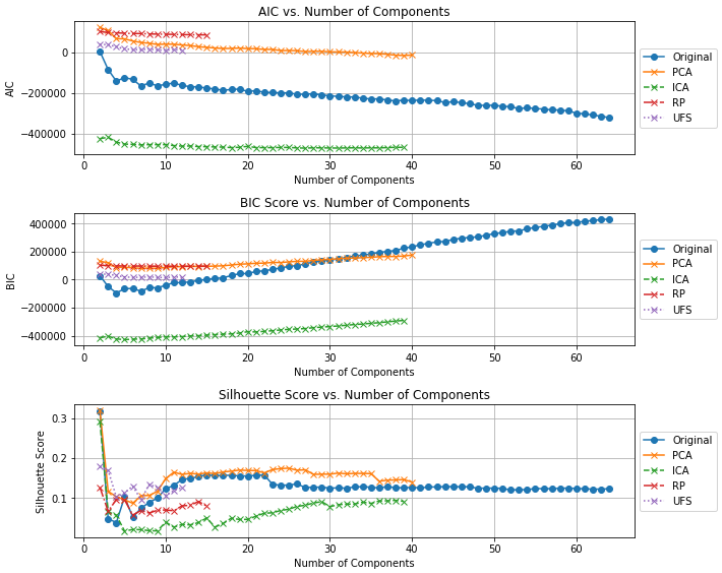
For the digits data base using k-means clustering after dimensionality reduction, the UFS based method produced the best results when a cluster size of 10 was used. With a relatively low sum square error value, the fastest time, and the highest silhouette score, it was the clear winner as shown by the benchmark testing below.

```
--------------------------------------------------------------------------------
*  K-Means Digits Benchmark  *
--------------------------------------------------------------------------------
init            time    inertia homog   compl   v-meas  ARI     AMI     silhouette
________________________________________________________________________________
Original        0.18s   69664   0.671   0.712   0.691   0.558   0.688   0.145
PCA-based       0.16s   64058   0.604   0.652   0.627   0.467   0.623   0.161
ICA-based       0.19s   31      0.552   0.665   0.603   0.419   0.599   0.018
RP-based        0.16s   63980   0.388   0.417   0.402   0.297   0.396   0.122
UFS-based       0.12s   8889    0.713   0.716   0.714   0.672   0.711   0.217
```

K-means of Digits Using Dimensionality Reduction



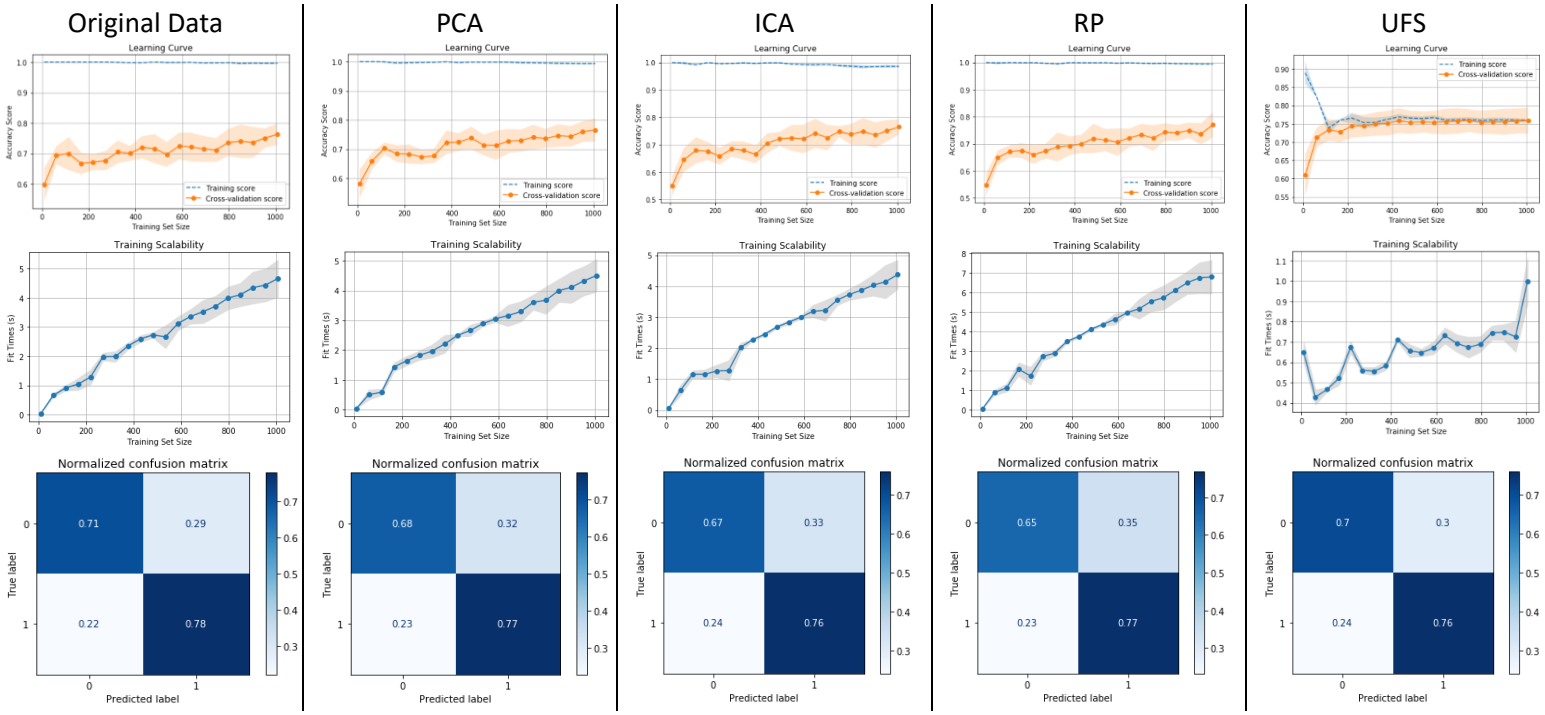
EM of Digits Quality Using Dimensionality Reduction




```
--------------------------------------------------------------------------------
*  EM Digits Benchmark  *
--------------------------------------------------------------------------------
init            time    aic     bic     homog   compl   v-meas  ARI     AMI     silhouette
________________________________________________________________________________________
Original        0.94s   -237537 233831  0.879   0.590   0.706   0.425   0.696   0.127
PCA-based       0.33s   -13549  175653  0.859   0.585   0.696   0.426   0.685   0.140
ICA-based       0.43s   -467138 -286945 0.821   0.560   0.666   0.411   0.654   0.094
RP-based        0.72s   80484   110365  0.687   0.463   0.553   0.303   0.537   0.086
UFS-based       0.74s   -1165   18826   0.698   0.500   0.583   0.345   0.565   0.015
```

When varying the number of clusters, it was less clear to pick an elbow in the charts above at what the ideal amount would be. Judging by the expectation maximization charts, it looks to be the case that above 20 or so features, the benefit seems to fall off.

Neural Network Learner + Dimensionality Reduction



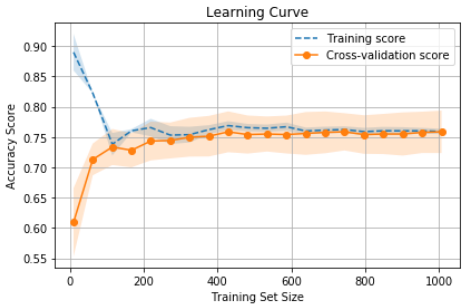
DATASET	ORIGINAL	PCA	ICA	RP	UFS
SCORE	0.771	0.773	0.767	0.770	0.758
TIME (SEC)	62.48	57.01	56.55	60.20	46.47
HIDDEN LAYERS	(20,20)	(20,20)	(20,20)	(40,20)	(10,10)
SOLVER	*lbfgs*	*lbfgs*	*lbfgs*	*lbfgs*	*sgd*
LEARNING RATE	*Constant*	*Constant*	*Constant*	*Constant*	*Constant*

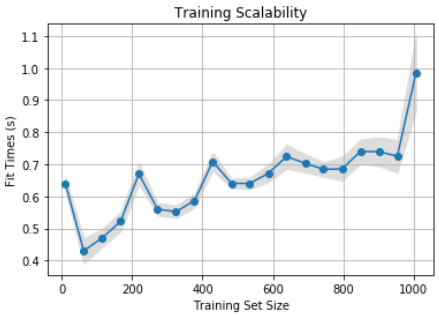
The above charts are the result of using dimensionality reduction on the wine quality dataset to reduce the amount data needed to train a neural network. For all cases but the UFS, the training error was very small for all sizes of training sets. This suggests a large amount of variance which can be seen in the accuracy of the test set. For all cases, the accuracy involved does not seem to vary much. The only meaningful difference shown is in the amount of time taken to train with the data provided after UFS, which can be seen in the Training Scalability plot as well as the total time taken to train in the table above. This is likely because the number of features was reduced from 11 to 4. For the other cases, the dataset was only marginally shrunk down to 8 or 9, which didn't manifest in a large decrease in the amount of time needed to train the neural network, although it is presumably a more generalized model.

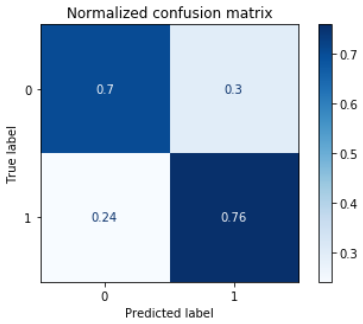
In each case, the test data set continued to increase in accuracy with an increase in the training set size. This implies that maybe a larger dataset would continue to improve the accuracy of the model and ensure that the model was not overfitting the data.

Neural Network Learner + Dimensionality Reduction and Clustering

K-means



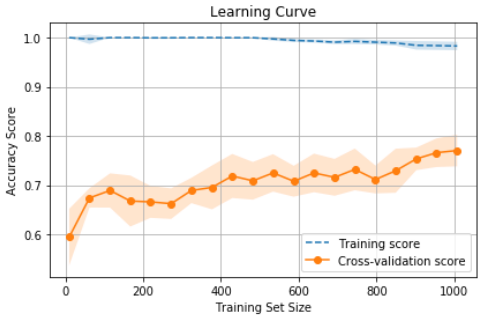


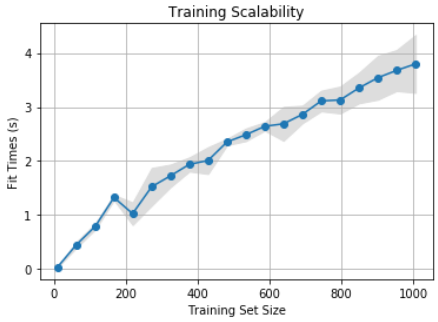


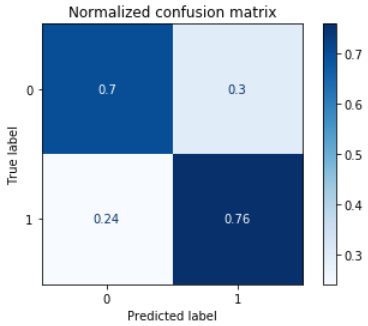
When using k-means to generate cluster labels to use as features for the wine quality dataset, not much improvement was observed in the accuracy of the model. With adding on one extra feature, the time needed to train the model was not very noticeable.

When comparing the cluster labels to the original wine labels, only 938 out of 1599 were correct, or about 59%. This most likely results in a feature that adds little value, hence the unnoticeable result in adding it.

Expectation Maximization







When using Expectation Maximization to generate cluster labels to use as features for the wine quality dataset, still no meaningful performance was gained. As seen before when using dimensionality reduction, the training set seems to perform very well, but the test set still falls around the 0.77 mark. With the performance continuing to increase with the training set size, it seems that more data would aid in improving the accuracy of the model.

When comparing the Expectation Maximization cluster labels to the original labels, even fewer were correct than the k-means case, only getting 736/1599 correct or 46%. Again, this most likely results in the feature not even being used amongst the other more meaningful parameters.