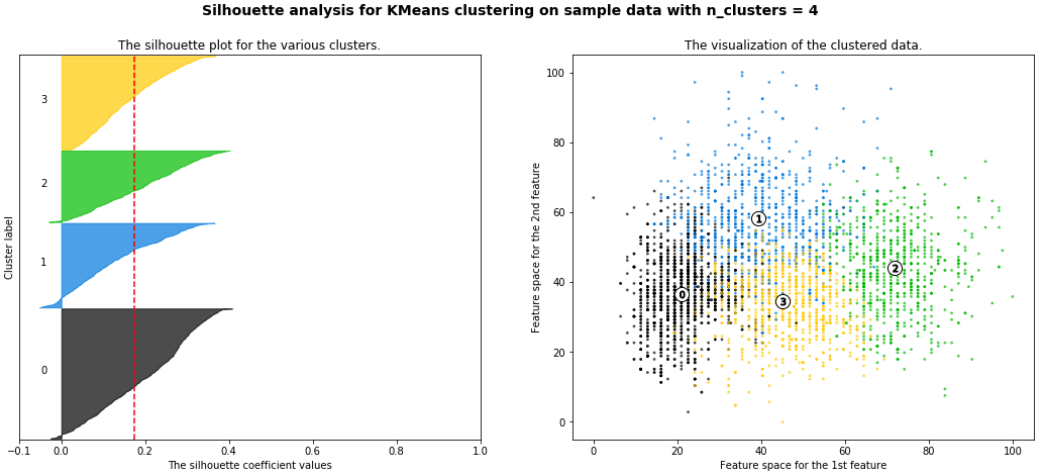
Data set 1:

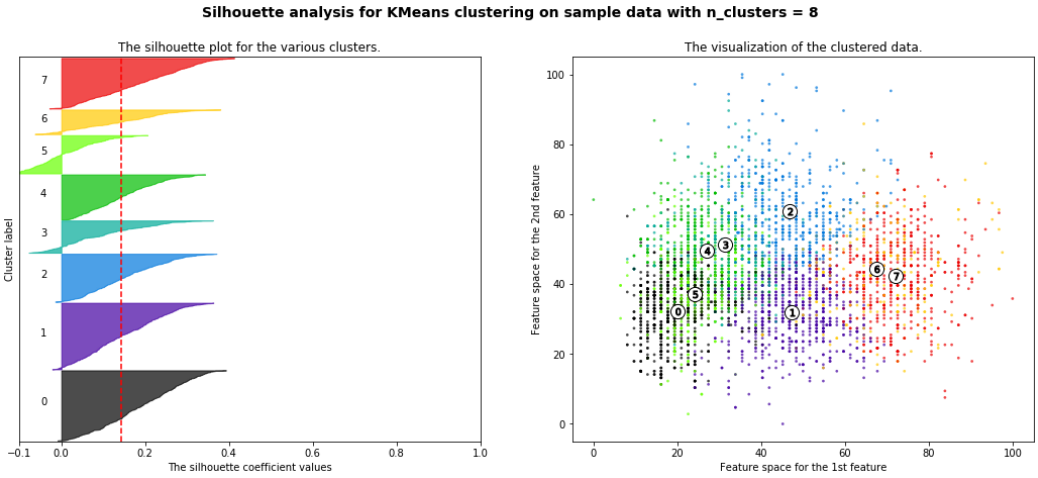
I chose to use the same wine quality data set as in assignment 1. One reason is that most previous supervised learning methods performed not very well against this data set. I'm wondering maybe I need to do a feature selection or feature transformation to help get better classification results.

1. Run the clustering algorithms on the data sets.

Because my wine quality data set have 8 classes, I chose to test K means for K = 2, 4, 6, 8 for simplicity. And in order to see compare the performance of different K, I plotted the silhouette coefficients for every run. The initial clustering result of my raw data seems to depend on one feature *free.sulfur.dioxide*, which has a significant larger range of values than all other features. Then I realized that the Euclidean distance used by sklearn was dominated by this large variance feature. So, I chose to use a min max function to normalize all my feature and rerun the algorithm.



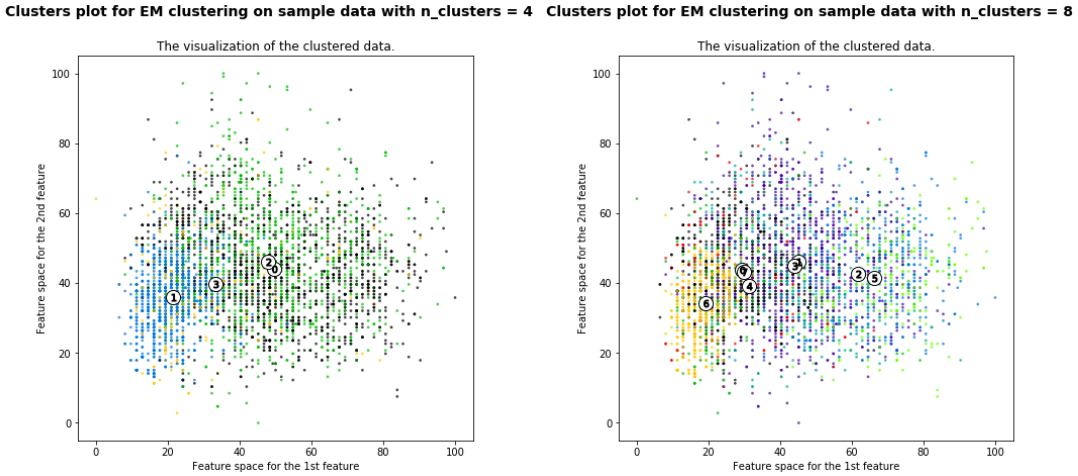
The result clusters can separate very well by several different features, and I plotted the two features that separate the clusters best. After comparing K=4 and K=8, it's obvious that K=8 only divide the data into more subgroups but can't really provide meaningful separations. As you can see in the following scatter plot, clusters [3, 4], [5, 0], [6, 7] are subgroups of clusters in the above scatter plot.



Both Silhouette Coefficient and Normalized Mutual Information (NMI) were calculated to test the performance of different K. It seems that NMI score is highest when K=2, while its value is only around 0.1. All other Ks have NMI score of 0.08 to 0.09. These NMI values show that our clustering results is very poor. I believe it must have been caused by treating all 11 features equally important. And because I normalized the feature so that all of them can influence the distance a lot, some features that don't really distinguish between wine qualities introduced a lot of variance to my clustering. That's will lead

me to the following sections of feature selection process. Also, the Silhouette scores were significantly lowered due to the same reason.

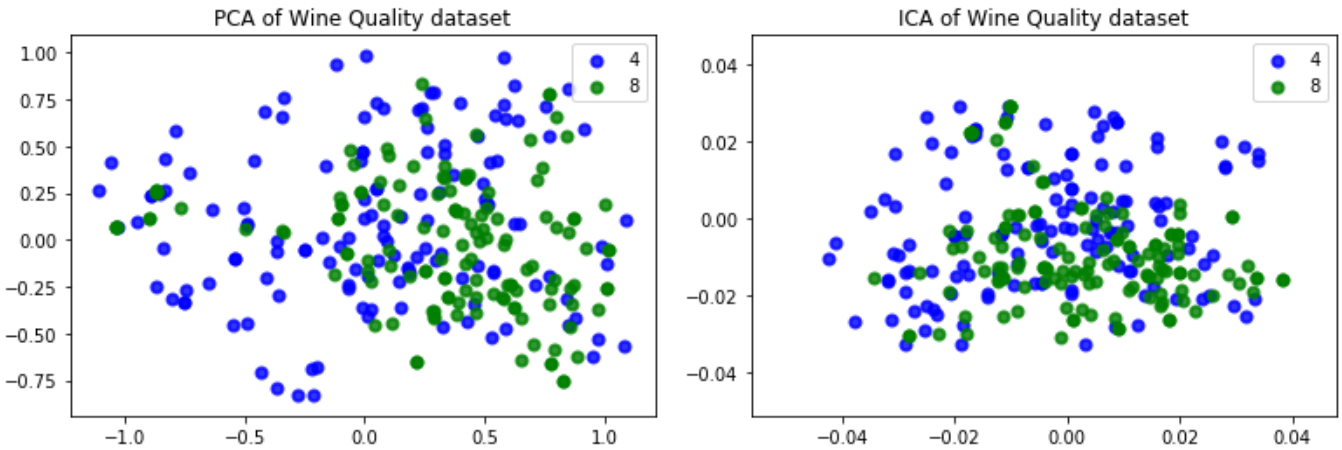
As for Expectation Maximization clustering, I use the same normalized values for all features and use the same set of components [2,4,6,8]. The output of n_components of 4 and 8 are:



And the NMI score of these two results are 0.056 and 0.078 respectively. Just by looking at the NMI scores, it seems EM performed even worse than K means. In addition to the two features plotted above, I also plotted all other 9 features as to see how well those features can separate the clusters. There seems to be 4 features that can provide some extra information for the clusters. While the other 7 features didn't divide the clusters at all. So, I think it will make a lot of sense to filter some features and redo the clustering. I also noticed that the min max normalization can't deal with features that have extreme values, the distribution of normalized values will be distorted towards one side only. So it may be useful to try some other normalization methods like quantile normalization.

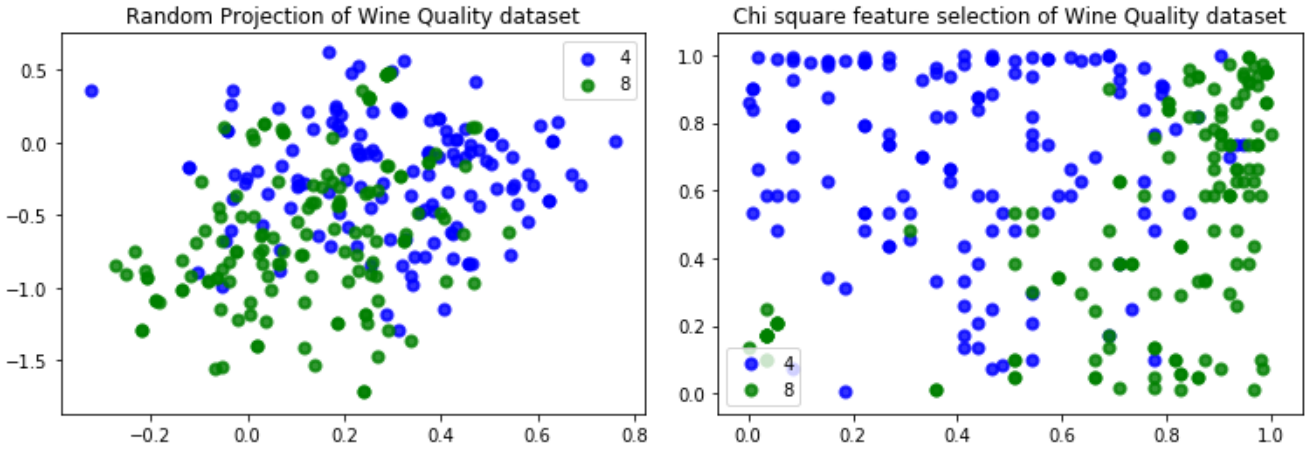
2. Apply the dimensionality reduction algorithms to the two datasets.

I first did PCA and ICA to my wine quality data set. I also tried to use the min max normalized data, but it doesn't change the separation of points for both algorithms. It's understandable since the variability of samples were kept the same while only the magnitudes of values were changed, so the percentage of variance explained by each component is the same. As far as I can see in the following plot of wine samples with quality 4 and 8, the PCA result can separate them much better than ICA. The amount of variance explained by PC1 and PC2 are 31.76% and 14.22%, which are pretty high given such a complex data set.



But if I plot the samples with other larger wine qualities, the scatter plots look messy. The classes of 4, 5, 6 can't be separate by the components from both methods. This seems to show that doing feature transformation can't just separate the classes directly, but ideally the transformed values can provide better classification results when applying a feasible

classifier. I then tried to use the random projection method for feature transformation. I didn't see any difference after running RP several times, maybe that's because I set the random state of 10 for all the runs. I also chose to use a univariate feature selection method of selecting the 5 best features base on a score function. This method appeals to me because I already found there are 4-5 features have better separation of the classes, I wonders if this algorithm can identify those features. And I also tried several score functions includes Chi Squared test and mutual information.

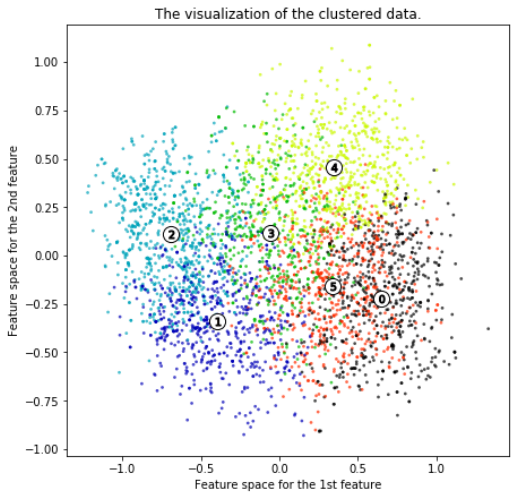


Just by looking at the RCA and feature selection results, the Chi Squared feature selection provided a very clear separation of the samples. And the random projection also provided a slightly better separation than the other two feature transformation methods. To compare the feature selection and transformation methods is not fair because the known classes information was used in the feature selection. The additional information will help with the feature selection, and I believe the selected and transformed features can do better in classification. But this supervised method also require me to check for overfitting problem, that's a big drawback of these supervised feature selection methods.

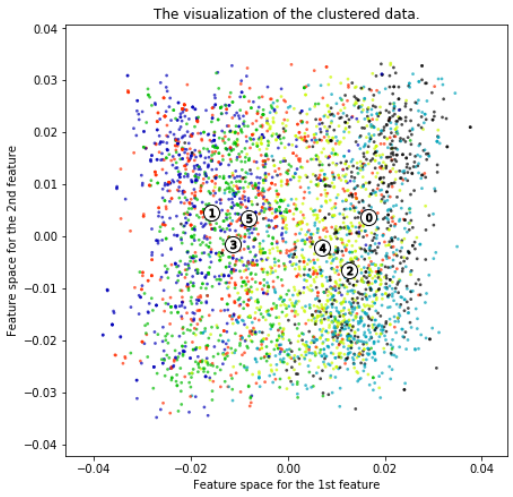
3. Reproduce your clustering experiments

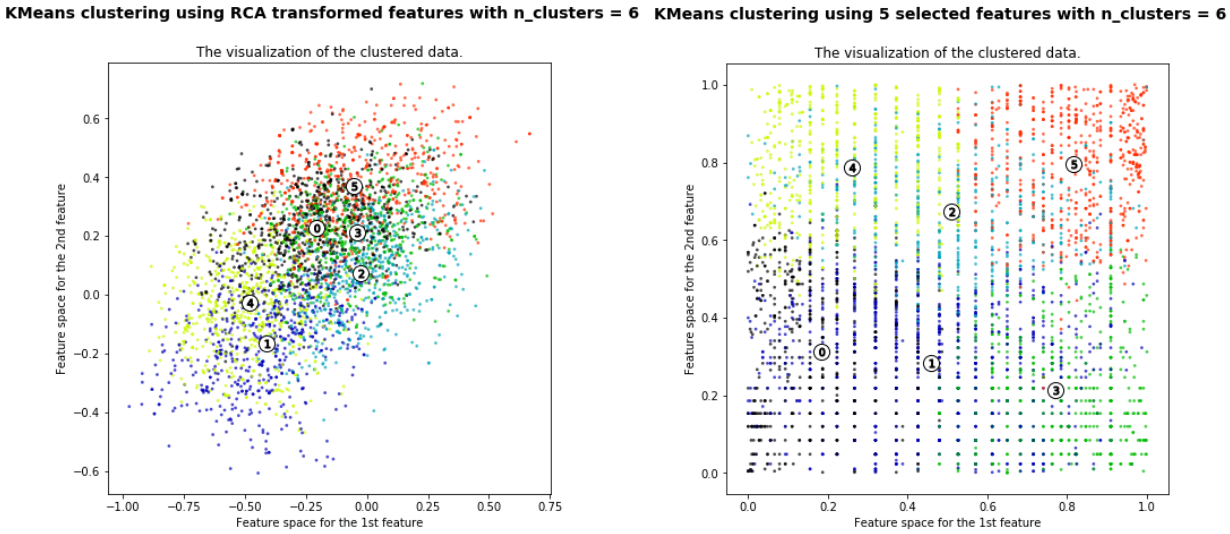
I redid the clustering use my results of 4 feature transformation methods. For K means clustering, both features set works best on K=6 setting. And the mutual information scores for the following 4 methods are 0.073, 0.088, 0.058, 0.071. The NMI scores did improve comparing to the raw feature values, but not much improvement when comparing to simple min max normalized features.

KMeans clustering using PCA feature transformation with n_clusters = 6



KMeans clustering using ICA transformed features with n_clusters = 6



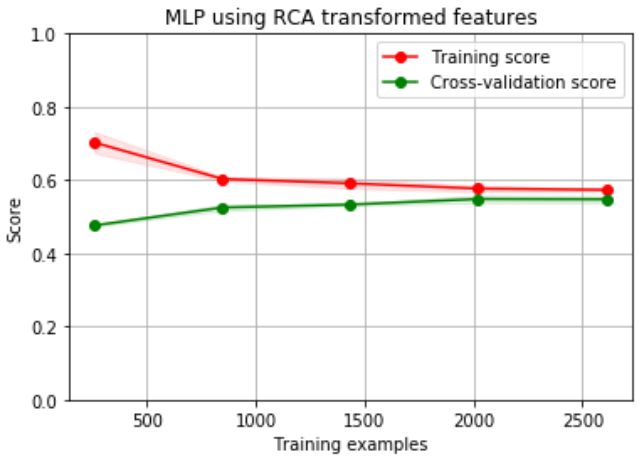


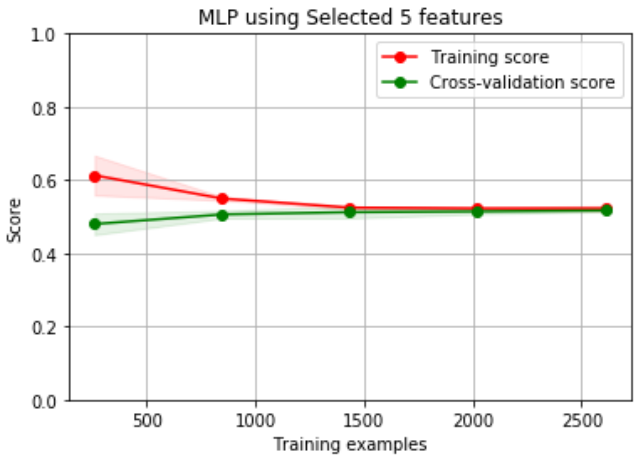
The separation results we can see in the scatter plots can't be used as a deterministic evaluation of the clustering results. The ICA clustering looks not very distinctive, but it has a much higher NMI score than the selected 5 features. To compare the results with previous clustering, I want to say all 4 methods took care of the extreme values in the features and can provide a more unbiased clustering than the min max normalized result.

As for EM clustering for the 4 methods. The NMI score of 4 transformed data set are 0.074, 0.092, 0.052, 0.074. The ICA transformed data get the best mutual information score again. Since I already controlled the random state and number of components used in each method. I'd like to say that ICA provided some very good independent components space for all the clustering methods. I think the reason why ICA perform better is due to the independence relation between the features, ICA can extract that independent information better than other ranking based methods.

4. Rerun your neural network learner.

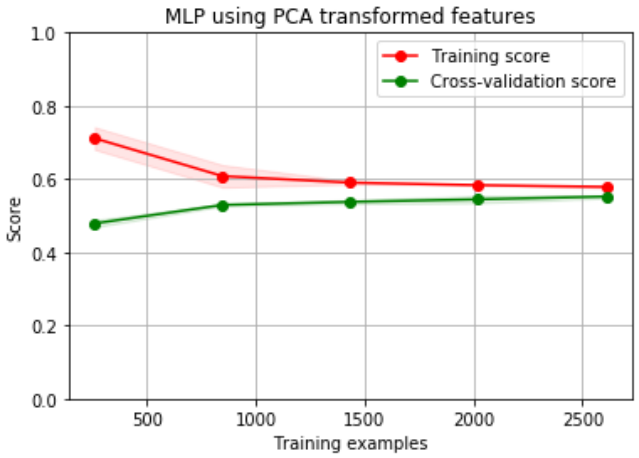
Using the 4 transformed data set on my previous optimal ANN model (20,5 hidden layer setup). The PCA and RCA transformed data suffered a little bit on overfitting as the final training score is about 0.03 higher than CV score. Though the training score of different data set are different, the CV score is very similar at around 0.55. Comparing with my previous ANN result of 0.46 optimal accuracy, these results show significant improvements. My understanding is that ANN can't handle the extreme values of that one feature in my previous runs, that reasonable because some extreme value will dominate the activation function given some similar weights. The feature transformation and selection methods can deal with this problem easily.

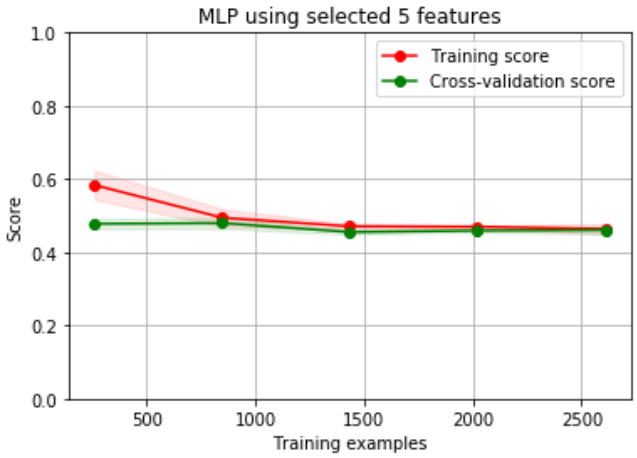




5. Rerun your neural network learner use clusters as classes.

The K means clustering results from each transformed data set were added to the features data frame. And the same ANN structure was used to train different models. The result of ICA was the same and converge at the same CV score. And the overfitting situations of PCA and RCA results were not as bad as before. But the result of selected 5 features data is not as good (0.48 CV accuracy). That is also related to the absolute values of the classes, it's much larger than selected and transformed 5 features. The overall benefit of adding the K clusters as a feature is obvious. I think that's due to the added complexity of this feature. The cluster numbers were derived from the original features, but since the clustering process is not perfect, it includes some random information that may help a complex neural network perform better.





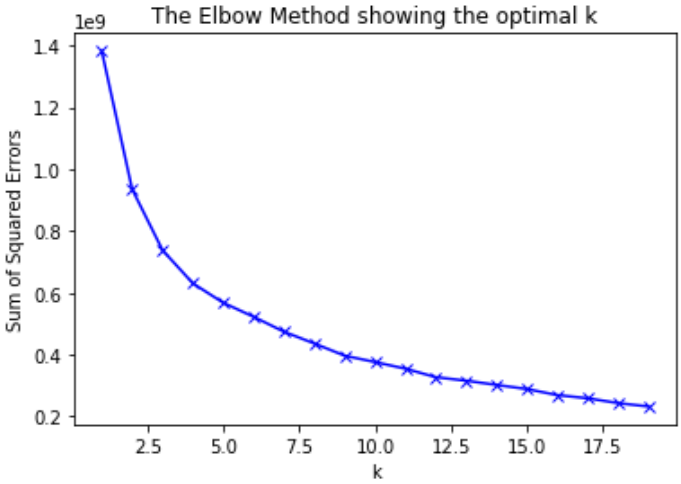
Data set 2:

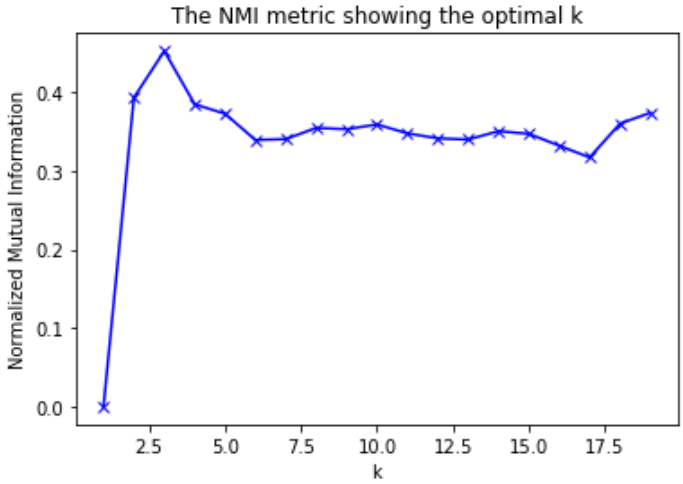
I found the Titanic survival data set I used in Assignment 1 can't be visualized in the clustering scatter plots because the raw data are all integers. So I chose to use my own data of some biological gene expression data set. The data set contains normalized expression values of 5091 genes from 76 samples. The samples came from 5 different cell types of health control or patients with SLE disease. I'd like to see how well the patients cluster together and the effect of different cell types.

1. Run the clustering algorithms on the data sets.

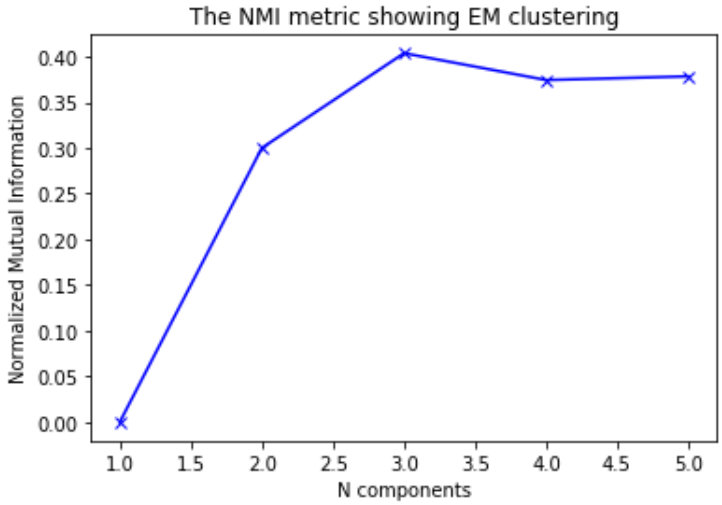
I used the Diease/Health feature as the classes (y). Then there are 34 health samples and 42 disease samples.

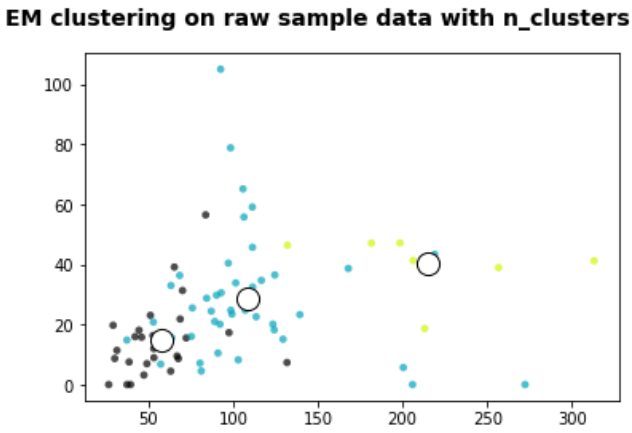
Because there are too many features to be plotted in a scatter plot, I chose to plot my metrics of evaluating the clustering instead. Using the Elbow method of plot within cluster variances, I can see a huge drop of variance between K equals 1 to 4, while the variance didn't decrease so much after 5. And by plotting the normalized mutual information, I can see the NMI reaches a peak at 3 and dropped from 4 and beyond. So a K of 3 or 4 seems to be an optimal choice for this data set.

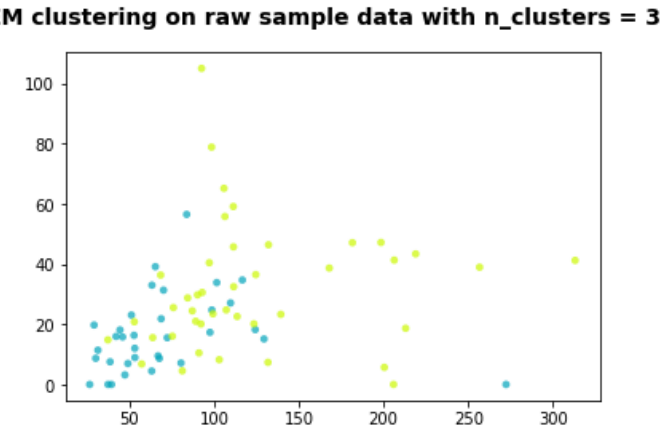




For EM clustering, there seems to be a significant run time increase due to dimensionality curse. A feature space of 5000 features will need at lease quadratic more time than a feature space of 11 features. That's why I only tried to select 2 to 6 components for EM, and the NMI score of those clusters are the following. Also, because the size of features space is too big, I just iterated through some features to get a reasonable scatter plot of the EM clusters. Just by looking at the two genes plotted here, we can see a clear separation of the samples.



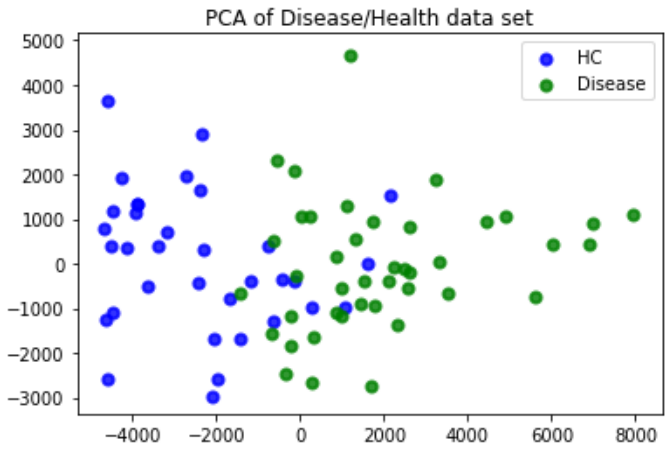


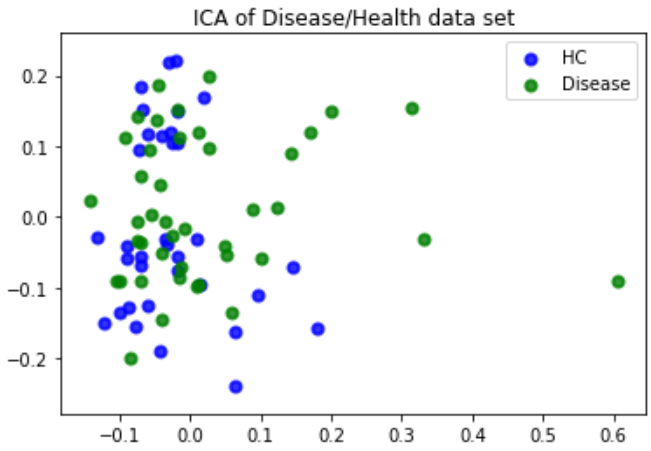


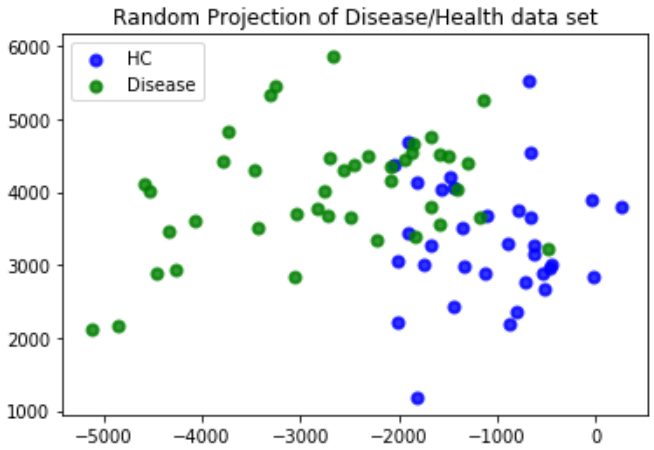
The left plot shows the 3 clusters formed by EM, while the right plot shows the two classes of Health/Disease patients. It's quite obvious that my two EM clusters to the right represent the Disease samples and the other cluster is health controls. To this extent, although the NMI score of the above clusters is around 0.4, it's actually separate my classes very well.

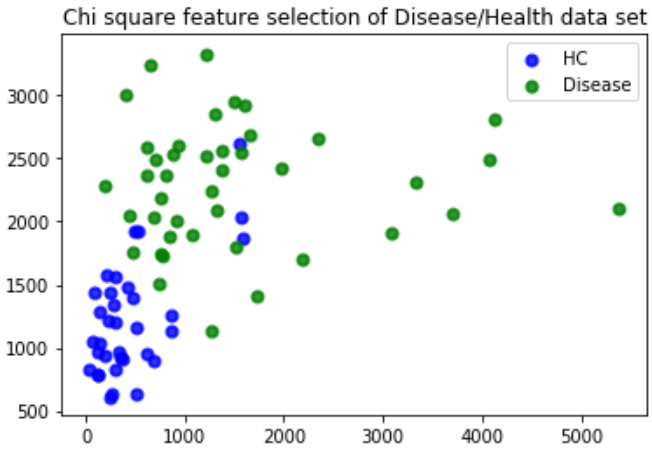
2. Apply the dimensionality reduction algorithms to the two datasets.

For simplicity reason, I just tried to reduce the dimension of my data set to 10 using the 4 algorithms. All algorithms performed very well except for ICA. The other three methods can extract components that accounts for significant amount of variance (49.9% and 11.7% for PCA top 2 components) or basically the top list of differentially expressed genes I got before (feature selection results). As for ICA, it's very possible that I didn't choose two good enough components to show in the scatter plot, but based on my previous results, the ICA components can perform very well. So, no worries for ICA.



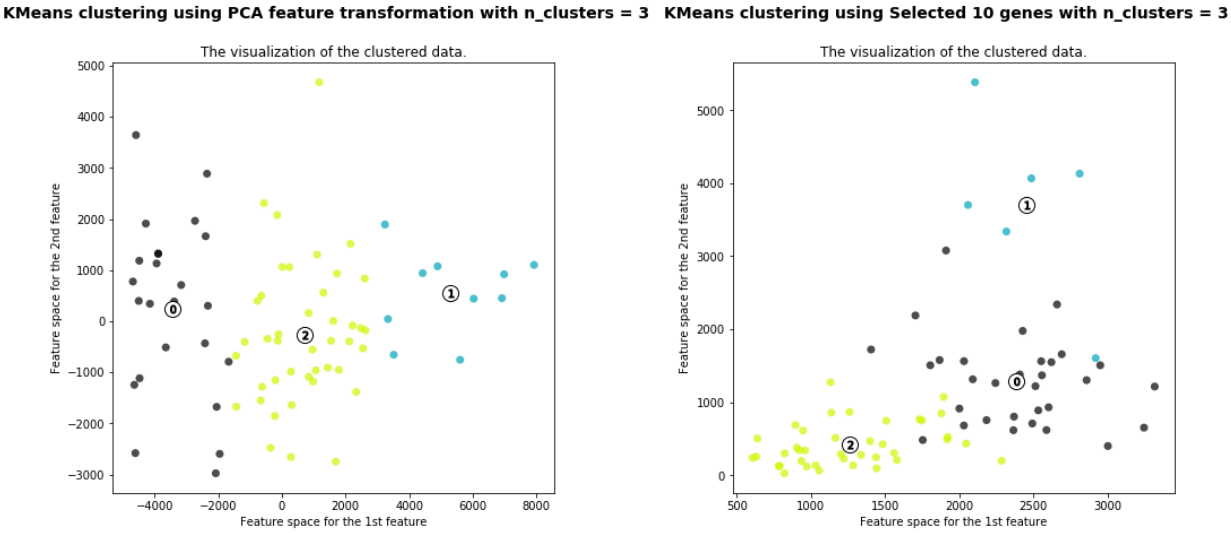






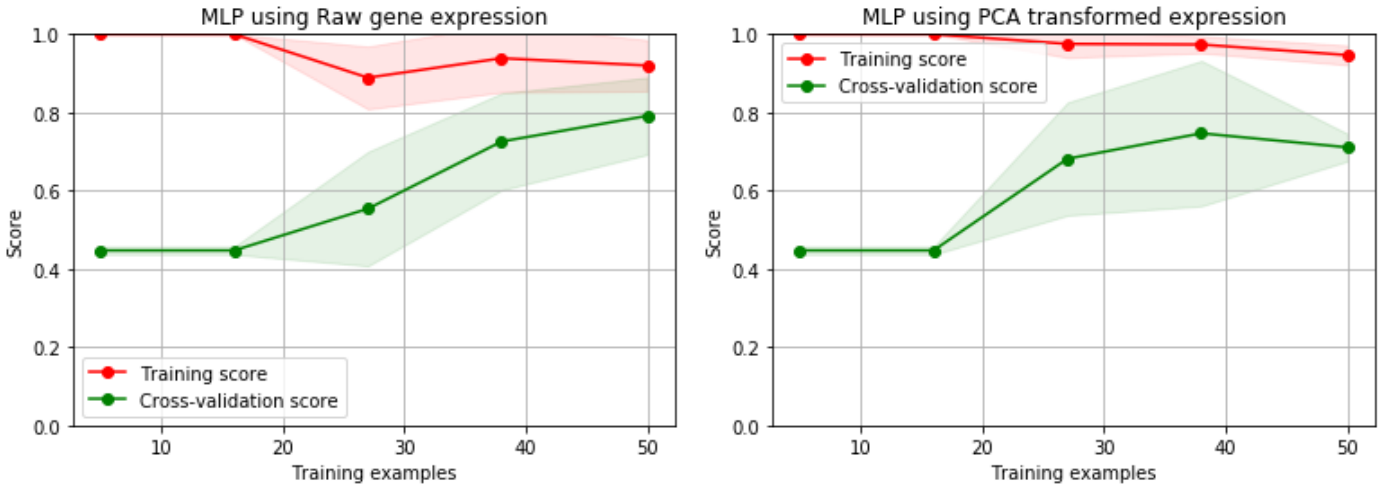
3. Reproduce your clustering experiments

All methods have a peak NMI score at n_clusters equal 3. The NMI score is around 0.37 using the PCA and Feature selected data set. These scores are also very high comparing my previous results. The 3 clusters predicted here can cover the actual 2 classes very well.

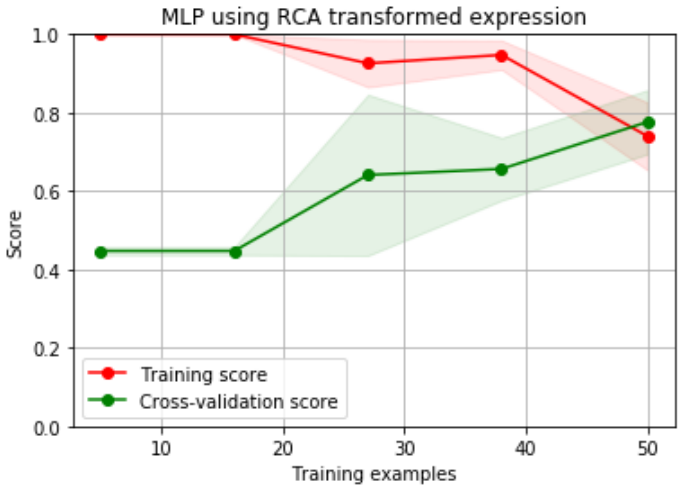


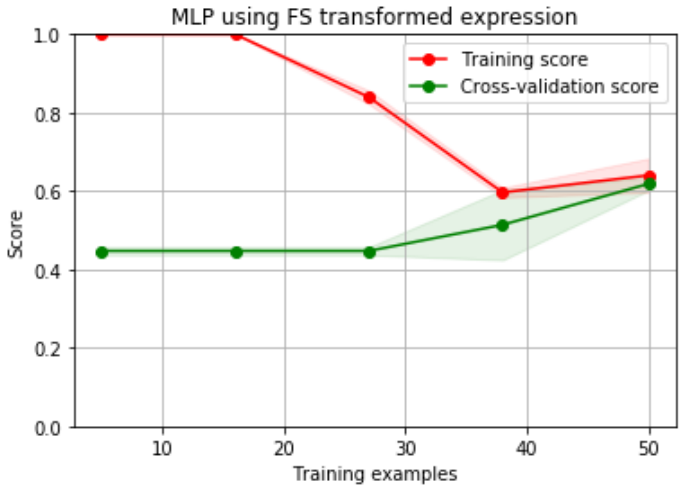
4. Rerun your neural network learner.

Because this is a new data set, I'll run the same neural network learner on raw features data and transformed features. Since there are only 76 samples in my data set, the learning curves didn't make much sense. But the final cross validation scores can evaluate the performance very well. For the 4 transformed data set, it actually reduced the performance of ANN. The raw gene expression data have an even higher prediction accuracy. That should be a consequence of dimension reduction algorithms, I can get much fast model convergence when using 10 transformed features. But the accuracy can't compete with a model using all 5090 features.



If we just compared the end performance of the 4 feature transformation algorithms. The RCA transformed data achieve a 0.79 prediction accuracy using the full data set. And the feature selection method didn't work well because it reduce feature space to 10 but didn't combine the information of different features. The other components extraction based methods seem to fit better for such a large feature space, and FS works very well for smaller amounts of features.


MLP using RCA transformed expression
Score
Training examples
Training score
Cross-validation score
1.0
0.8
0.6
0.4
0.2
0.0
10
20
30
40
50


MLP using FS transformed expression
Score
Training examples
Training score
Cross-validation score
1.0
0.8
0.6
0.4
0.2
0.0
10
20
30
40
50