Assignment 3 :Unsupervised Learning and Dimensionality Reduction

In this report, I first used 2 clustering algorithms(kmeans clustering and expectation maximization) to cluster two datasets. Then, I applied the 4 dimensionality reduction algorithms to the 2 datasets and project the data to different dimensions. In the last, I ran the neural networks on one of the original datasets and the datasets I applied the dimensionality reduction algorithms on. In this part, I also treated the cluster label as a new feature and reran the neural network learner on the newly projected data. I re-used the two datasets in assignment 1 on this report.

Datasets Introduction

Dataset 1: Speech Emotion Recognition
Speech Emotion Recognition is the task of recognizing human emotion from speech regardless of the speech contents. We have been always wanting to achieve natural communication between humans and machines. However, we are still far from having a natural interaction between man and machine because the machine does not understand the emotional state of the speaker. That's why I want to pick the problem of speech emotion recognition(SER). It's important to let the machine have enough intelligence to understand human emotion by voice.

The rawest datasets I started with is a collection of 3 seconds audio files in which different actors repeat similar sentences with different emotions. Then I processed the audios with python audio analyzing library Librosa, and extracted three features:

MFCC (Mel Frequency Cepstral Coefficient, represents the short-term power spectrum of a sound).
Chroma (Pertains to the 12 different pitch classes).
Mel (Mel Spectrogram Frequency).

Each feature provides a lot of attributes, there are 180 attributes and 8 classes in total. To simplify the task I reduced the classes to 4 emotions that might be easier to identify: calm, happy, fearful, and disgust. The training size shrinks to 768.
Audio Data: https://drive.google.com/file/d/1wWsrN2Ep7x6lWqOXfr4rpKGYrJhWc8z7/view

Dataset 2: Forest Cover Type
Predicting forest cover type is learning the forest cover type classes and their cartographic variables, and predicting the cover type in the future. The forest in this study is wildness area with minimal human-caused disturbances. To do an inventory of natural resources helps to observe ecological changes and identify the type of biosphere within the area.

The original datasets have more than 580,000 instances and 54 attributes. To build the model efficiently, like what I did in assignment1, I randomly selected around 3000 instances to test.
Data Set: https://archive.ics.uci.edu/ml/datasets/Covertype

Part 1. Clustering

In the first part, I did clustering of two datasets with two algorithms: k-means clustering and Expected Maximization.

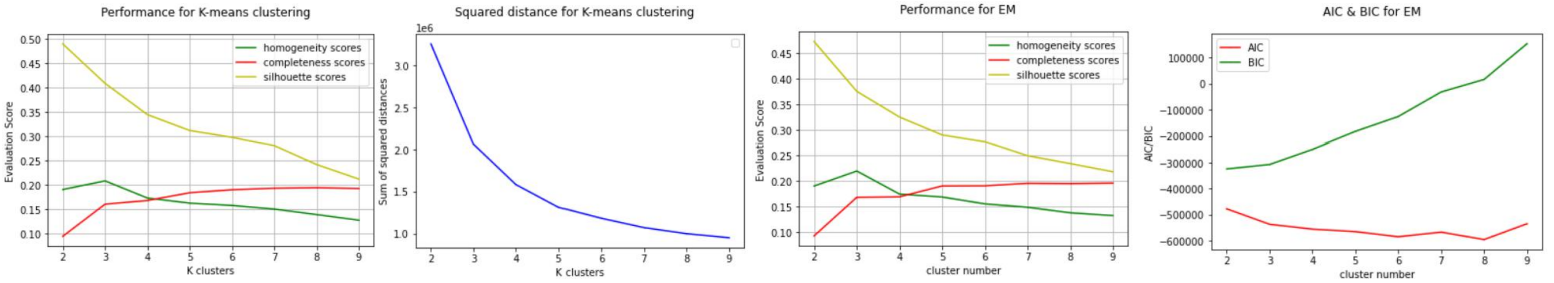
K-means clustering
K-means clustering is a simple and popular unsupervised machine learning algorithms. The K-means algorithm identifies k number of centroids, and allocates each data point to each cluster by reducing the in-cluster sum of squares.

Expectation Maximization (EM)
The expectation-maximization algorithm is an approach for performing maximum likelihood estimation in the presence of latent variables. It does this by first estimating the values for the latent variables, then optimizing the model, then repeating these two steps until convergence. It is an effective and general approach and is most commonly used for density estimation with missing data, such as clustering algorithms like the Gaussian Mixture Model.

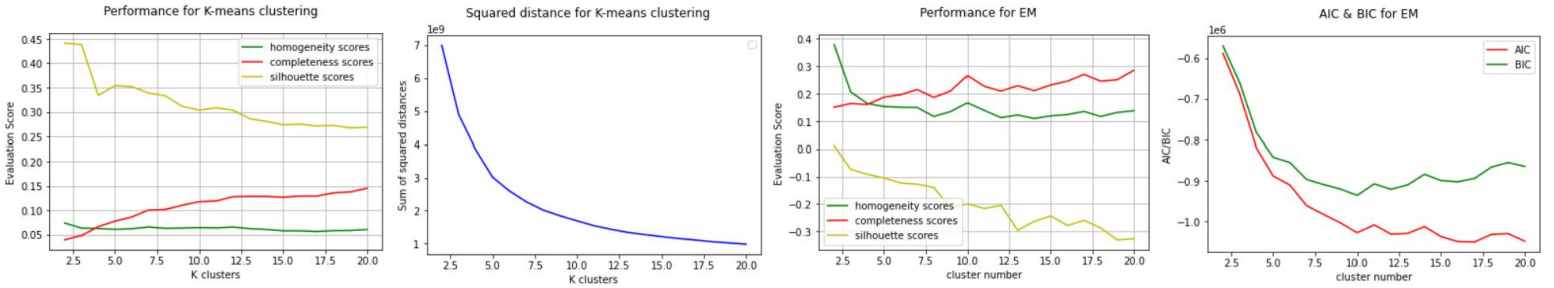
First, I want to introduce some evaluation metrics for Kmeans and EM. The evaluation metrics for kmeans include homogeneity (satisfies homogeneity if its clusters contain only data points which are members of a single class -- can have same labeled data in different clusters), completeness (satisfies completeness if all the data points that are members of a given class are elements of the same cluster -- can have differently labeled data in the same cluster), silhouette (how similar an object is to its own cluster (cohesion) compared to other clusters (separation)), the sum of variance (squared deviations from the mean of the cluster of all the observations belonging to that cluster). The evaluation metrics for EM include homogeneity, completeness, silhouette, and AIC/BIC(AIC means Akaike's Information Criteria and BIC means Bayesian Information Criteria. The AIC can be termed as a measure of the goodness of fit of any estimated statistical model. The BIC is a type of model selection among a class of parametric models with different numbers of parameters.)

First, I ran the analysis to find out the best K number for k means clustering. As the number of clusters goes up, the silhouette score and variance go down, as more clusters form the more similar data points to their cluster center. This can be explained as the data is so

complicated with 180 attributes and doesn't have clear separation, and more cluster centroids give lower variance. The homogeneity is highest when there are 3 clusters and goes down when there are more clusters. The completeness score keeps going up, when there are 8 clusters it stays stable and 7,8,9 clusters' completeness scores are very close to each other. It shows when 8 clusters more same labeled data are in the same clusters, and when there are 3 clusters each cluster tends to have same labeled data. It's logical with the dataset of 4 classes. Secondary, I ran the analysis for EM clustering on dataset 1, which shows the same pattern on homogeneity /completeness /silhouette, for BIC it keeps going up means the error variance goes up, the 2 clusters model has the lowest BIC value. But the AIC curve is going down and finds a lowest value when cluster is 8, means there is an estimated statistical model that finds 8 cluster that fits the model the best. Combine with the result from K means, 8 clusters would the best.



The same analysis was done to dataset 2(Forest Cover Type). In k means, the silhouette score goes down as more clusters formed but when k = 4 the curve shows a low point. The homogeneity score is somehow stable and the completeness score goes up as k goes up. In EM clustering, the overall silhouette score is lower than the one in K-means, there is lower variance in the clusters. The AIC and BIC curves show there are some optimal cluster numbers at 10 and 16, 10 cluster has the lowest BIC and 16 has the lowest AIC.



Part 2. Dimensionality reduction

For this part, firstly, I did dimensionality reduction of the two datasets with four different algorithms: Principal Component Analysis (PCA), Independent Components Analysis (ICA), Randomized Projections (RP), and Factor Analysis (FA).

Principal Component Analysis (PCA)

Principal component analysis (PCA) is the process of computing the principal components and using them to perform a change of basis on the data, sometimes using only the first few principal components and ignoring the rest. PCA is used in exploratory data analysis and for making predictive models. It is commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible.

Independent Components Analysis (ICA)

Independent component analysis (ICA) is a computational method for separating a multivariate signal into additive subcomponents. This is done by assuming that the subcomponents are, potentially, non-Gaussian signals and that they are statistically independent from each other. As an example, sound is usually a signal that is composed of the numerical addition, at each time t, of signals from several sources. The question then is whether it is possible to separate these contributing sources from the observed total signal. When the statistical independence assumption is correct, blind ICA separation of a mixed signal gives very good results.

Randomized Projections (RP)

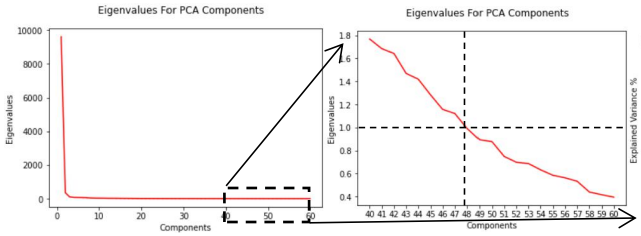
random projection is a technique used to reduce the dimensionality of a set of points which lie in Euclidean space. Random projection methods are known for their power, simplicity, and low error rates when compared to other methods. According to experimental results, random projection preserves distances well, but empirical results are sparse

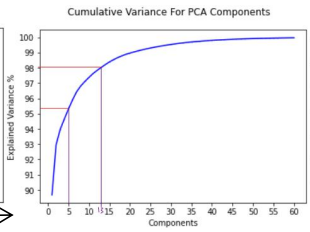
Factor Analysis (FA)

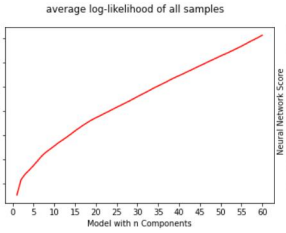
Factor analysis is a statistical method used to describe variability among observed, correlated variables in terms of a potentially lower number of unobserved variables called factors. For example, it is possible that variations in six observed variables mainly reflect the variations in two unobserved (underlying) variables. Factor analysis searches for such joint variations in response to unobserved latent variables.

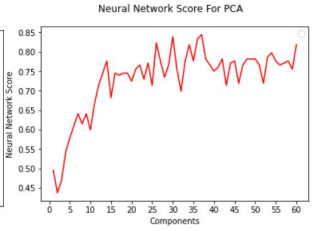
Dataset 1 - PCA

First I ran the PCA with 60 components(given that I have 180 features to start with) and got the eigenvalue of each component. As more components are added, the eigenvalue goes down. An eigenvalue of 1 means that the principal component would explain one variable's worth of the variability, so we can see from the chart when components get 47 it can explain all the variables. And more components wouldn't be necessary. The same thing with the cumulative explained variance, the total variance would be 100%, the first 5 components can explain 95% of the variables, and the first 13 components can explain 98% of variables.





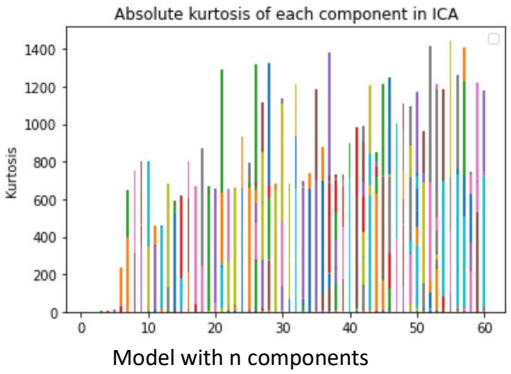


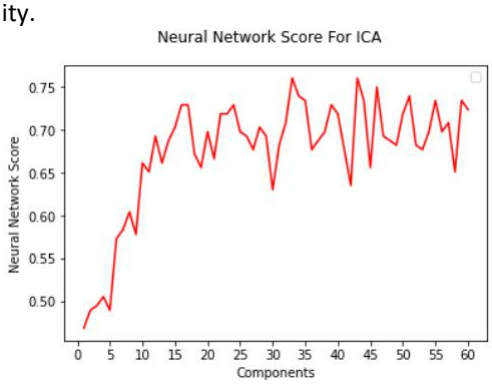


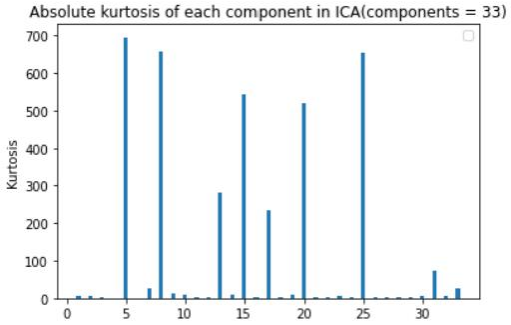
Then I use PCA to project data to lower dimensions with different components(from 1 to 60) and use the projected data to produce the neural network training and get the accuracy scores. The curve goes up and down but overall it's raising and somehow stable between 20 to 60. The accuracy is highest with 37 components and more components won't be necessary.

Dataset 1 - ICA

A necessary condition for ICA to work is that the signals be non-Gaussian(kurtosis). The kurtosis lower to 0 maximize the non-Gaussianity, so I did an analysis for absolute kurtosis for each component produced by different ICA with different component numbers(1 to 60). It's interesting to the projected data have some high kurtosis components and also very low components. Almost every projected data have some components that don't have low non-Gaussianity.



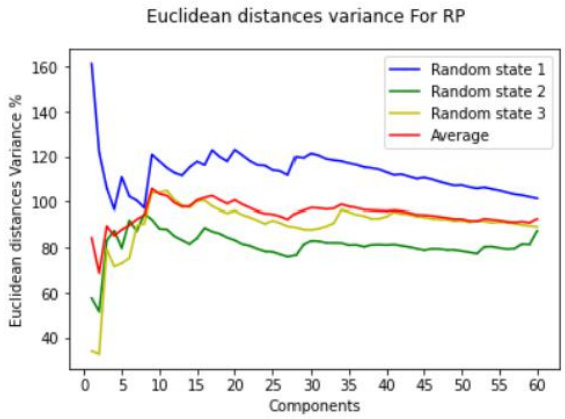


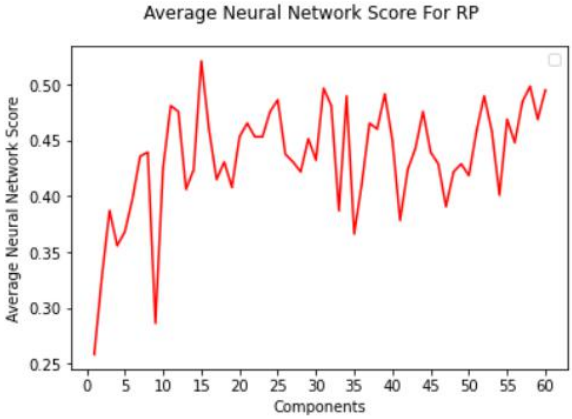


Then I ran the neural network learner on ICA with different components to get the accuracy score on different models. For 16 to 60 components ICA, they all have a chance to make neural network learners achieve high accuracy, which aligns with the high accuracy zone of PCA. The best accuracy is by ICA with 33 components. I did a kurtosis analysis of each component in 33-component ICA, it has more high kurtosis components than most of the other ICAs, I think it explains why it did the best.

Dataset 1 - Randomized Projection

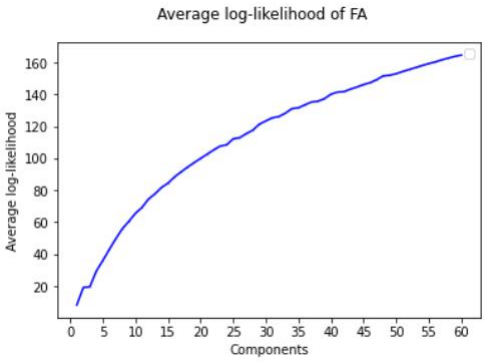
The randomized projection has its randomness, so I ran the algorithm 3 times for every analysis with 3 different random states. When there are only a few components it is more random, and when the number of the components gets bigger, it stays the same trend. The Euclidean distance variance shows the average variance raised as first till 9 components and slowly goes down. The neural network score shows more randomness than the other algorithms, it zig-zag crazily. But overall, for Randomize projection, the neural network accuracy is much lower than the neural network trained by data after PCA and ICA, which is not a very effective dimension reduction algorithm for this dataset. The highest accuracy component number is 15.

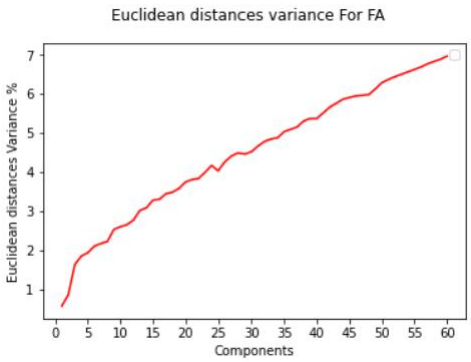


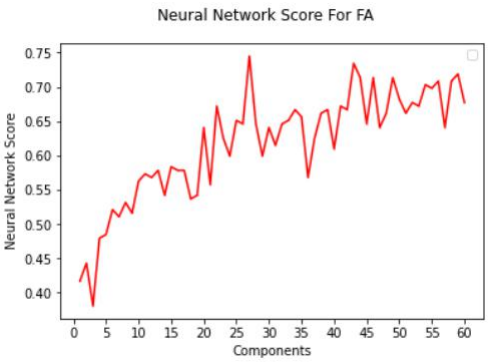


Dataset 1- FA

Factor analysis and PCA are somehow similar, they identify patterns in the correlations between variables. These patterns are used to infer the existence of underlying latent variables in the data. The average log-likelihood and euclidean distance variance are raising as more components appear, and for neural network learners, the trend of accuracy score is going up, but the highest score is by 27 components.



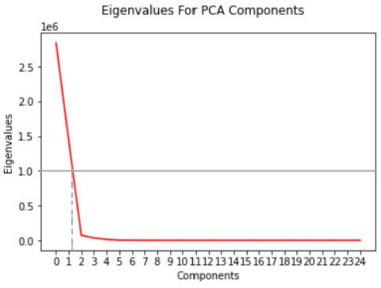


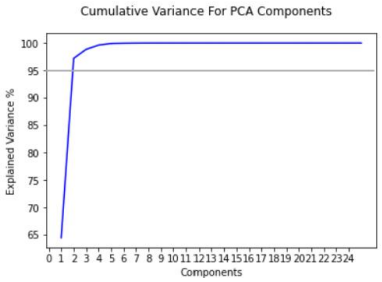


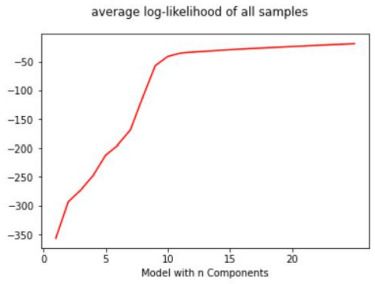
For dataset 2, I have done the same type of analysis. Due to the original feature number in dataset 2 being 54, the range of components I chose is 1 to 25.

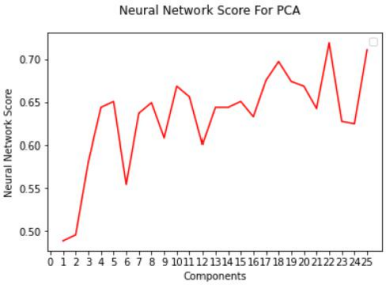
Dataset 2 - PCA

First I ran the eigenvalue analysis with 25 components PCA, the eigenvalue for each component starts from super high to super low(near 0), the first 2 components explain a lot of the variables(95%), and the other components are more and more trivial. On neural network learners, the best component number is 22.



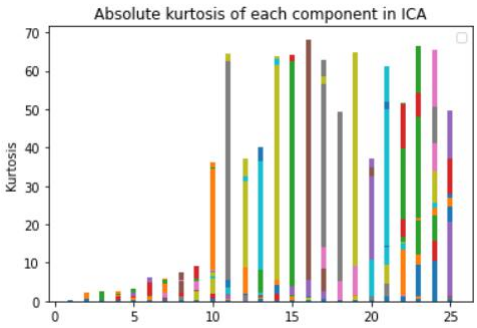


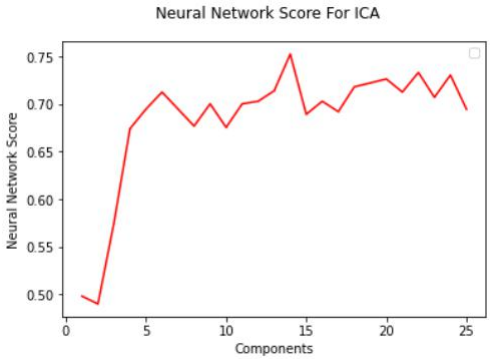


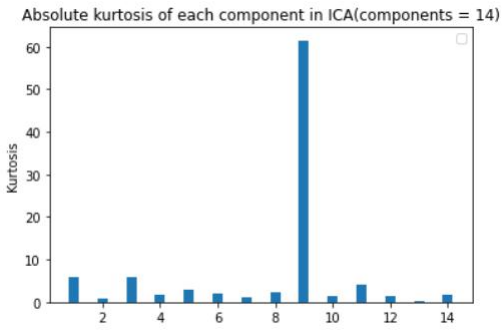


Dataset2 - ICA

The absolute kurtosis of each component in ICA shows different components ICA has all kinds of components' kurtosis value, some are very high but always combine with some low kurtosis components. As for the neural network score, the 14 components ICA got the highest accuracy. I rerun the kurtosis of each component on that one, this time we get only one very non-Gaussian component with some low non-Gaussian components.

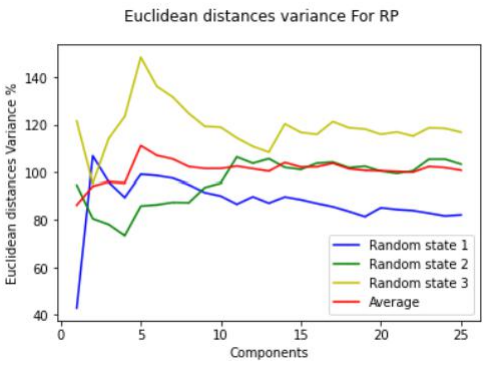


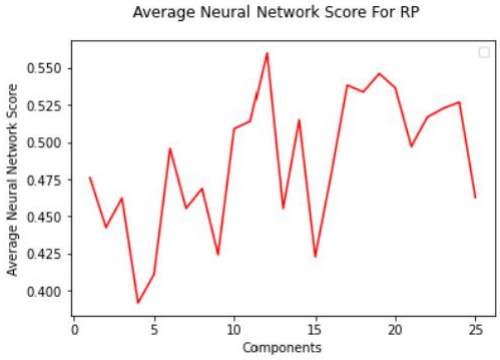




Dataset 2 - RP

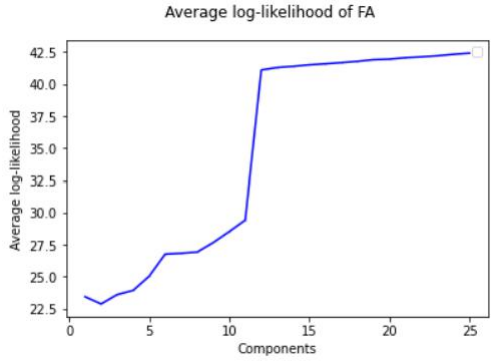
Randomized projection is still quite random, different random state leads to different variance. And average neural network score is somehow lower than ones in PCA and ICA, RP is efficient and cheap, but can't always have the best result. But still, it's much higher than the chance(12.5%). The best neural network score is by 12 components RP.

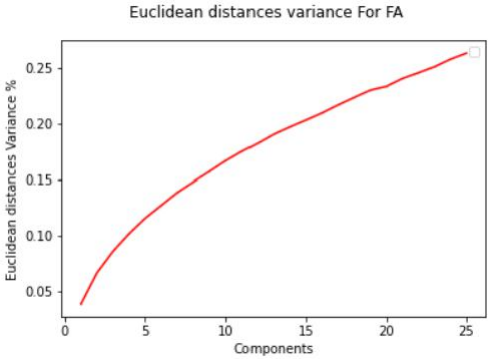


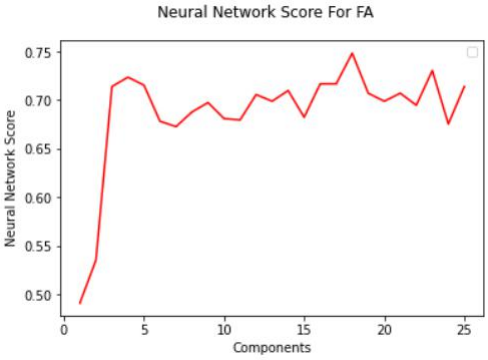


Dataset 2 - FA

For FA, it's analyzing the underlying factor from the first component, and when there are 3 components FA already reaches a high neural network score, it's highest on 18 components FA.





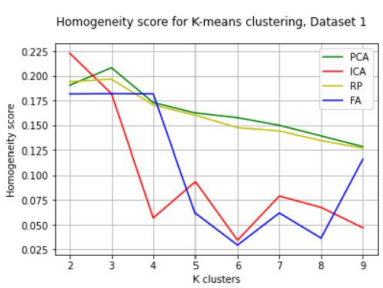


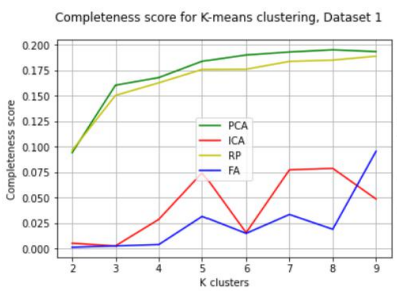
Part 3. Clustering after dimensionality reduction

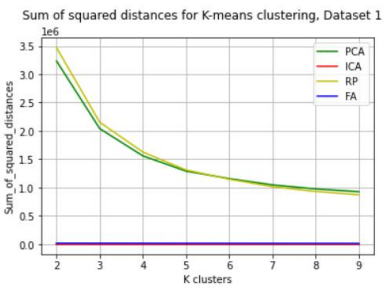
In this part, I explored the clustering of both datasets after four algorithms of dimensionality reduction. The two clustering algorithms I used are still means and EM. The number of components for each dimensionality reduction is picked from part 2, the optimal number of components by running neural network learner. Many performance evaluation metrics were plotted with various parameters. For kmeans, the metrics are homogeneity, completeness, silhouette, and variance score(sum of squared distance). For EM, the metrics are homogeneity, completeness, silhouette, AIC, and BIC. The changes of AIC and BIC curves for RP are hard to see sometimes, so I plotted a chart with only AIC/BIC curves for RP. (For each RP analysis, I ran 3 times to get the average.)

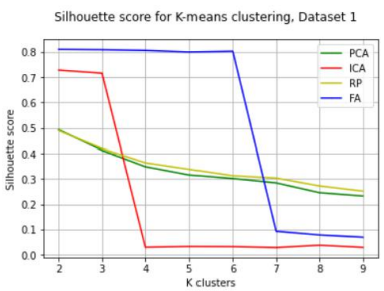
Dataset 1 - kmeans after 4 dimension reduction algorithm (components: PCA 37, ICA 33, RP 15, FA 27)

The first thing I noticed here is the similarity between ICA and FA. FA, PCA and ICA are somehow "related", but FA is a model of the measurement of a latent variable. This latent variable cannot be directly measured with a single variable. Instead, it is seen through the relationships it causes in a set of Y variables(Yn = bn*F + un). PCA and RP also shared some similarities, for each evaluation metric, their score didn't change much. It's nice that they show some consistency, which means the data projection didn't change much on how K-means clusters the data. The ICA and FA changed completely, the homogeneity and completeness scores are lower than what they were, but the silhouette score is much higher when the number of the cluster is limited, which means the projected data are more prone to be clustered. Combining all charts together, my observation is PCA and RP didn't change the clustering behavior much, and ICA shows the best clusters number is 2, while FA shows the best clusters number is 4. And overall FA has the highest silhouette score, also the cluster number equals the classes number. I'm looking forward to seeing how FA will be doing when I take clusters as new features.

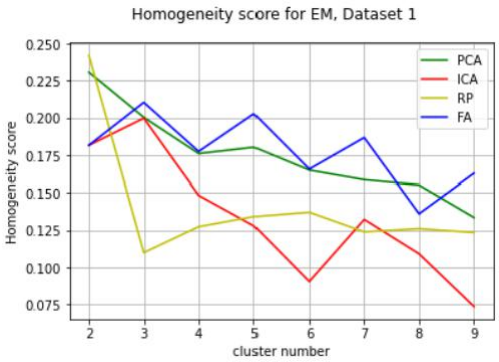


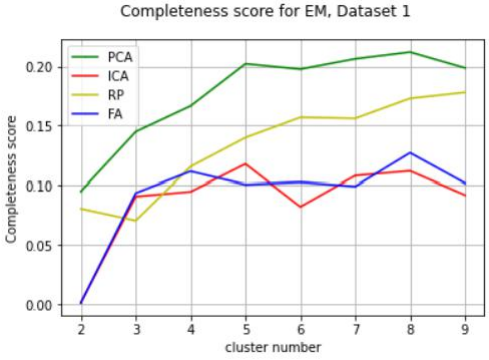


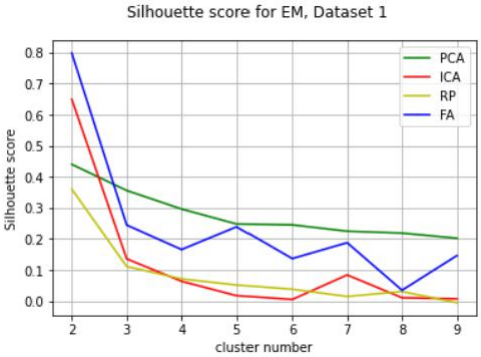


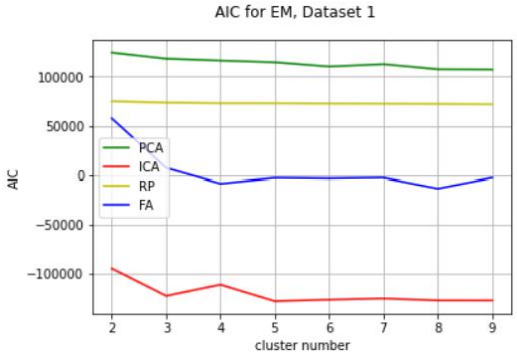


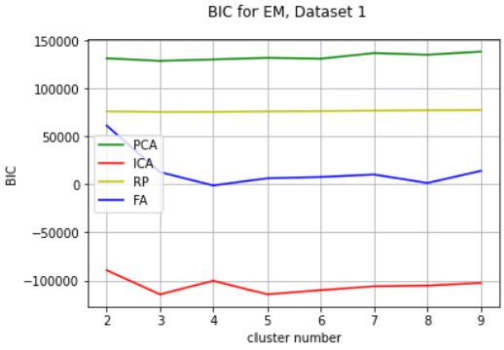
Dataset 1 - EM after 4 dimension reduction algorithm (components: PCA 37, ICA 33, RP 15, FA 27)
The clustering after EM looks very different, based on the completeness and homogeneity scores the PCA-EM clustering has the best performance, the PCA seems to have the same trend as RP. But for AIC and BIC, the performance order is ICA > FA > RP > PCA. Based on the model selection criteria, the ICA-EM model achieved the best clustering result.

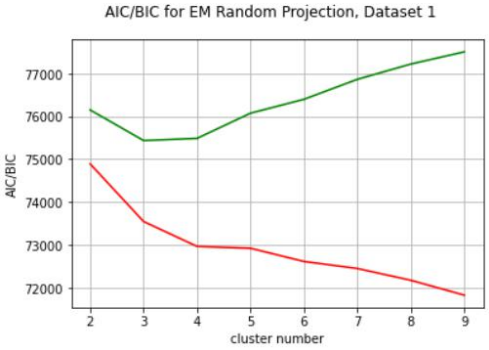




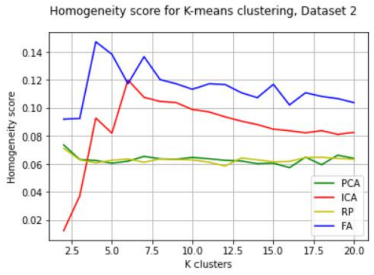


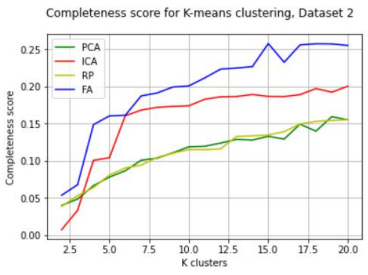


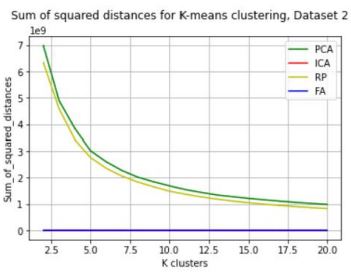


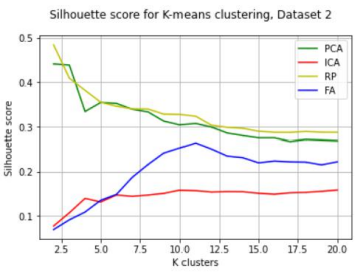


Dataset 2 - kmeans after 4 dimension reduction algorithms (components: PCA 22, ICA 14, RP 12, FA 18)
The results show the consistency between PCA and RP, as cluster number goes up, the homogeneity score of PCA and RP stay stable, the completeness keeps going up, and silhouette score slowly reduced, which is very alike to the K means clustering before dimension reduction but have higher scores, even though the PCA and RP didn't change the clustering pattern much but it makes the data more prone to cluster. ICA and FA changed the clustering pattern, the homogeneity and completeness scores are much higher than before, but the silhouette score is lower, which means the projected data can be clustered more related to their label but less similar to their own clusters.

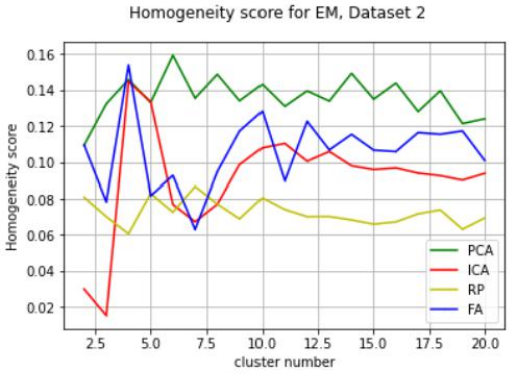


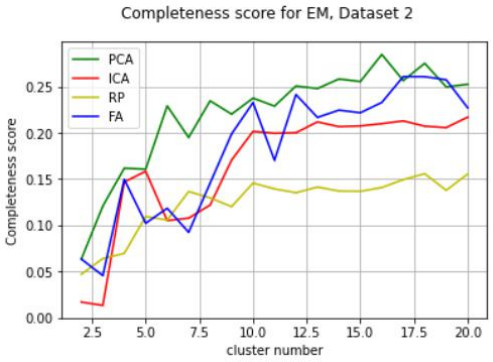


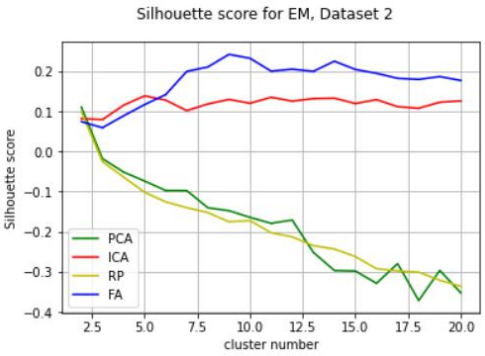


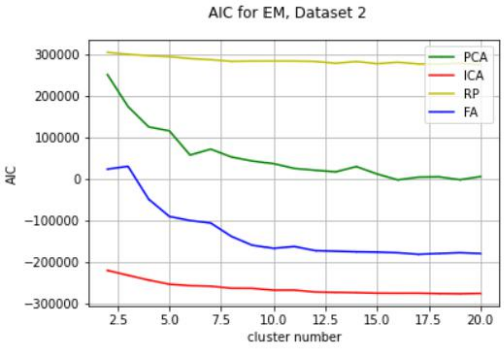


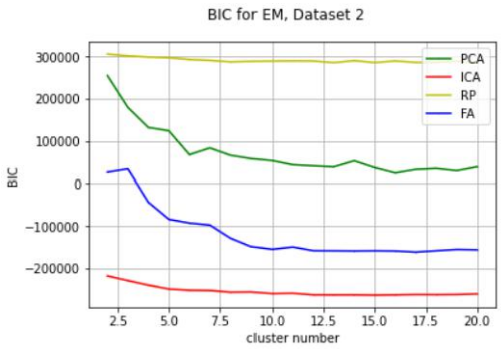
Dataset 2 - EM after 4 dimension reduction algorithms (components: PCA 22, ICA 14, RP 12, FA 18)
All four dimension reduction algorithms show different clustering patterns, their homogeneity and completeness score is close to the score before dimension reduction but slightly higher. The silhouette score for ICA and FA is much higher than the original EM clustering, also their AIC/BIC score is way lower than previous, which is preferred.

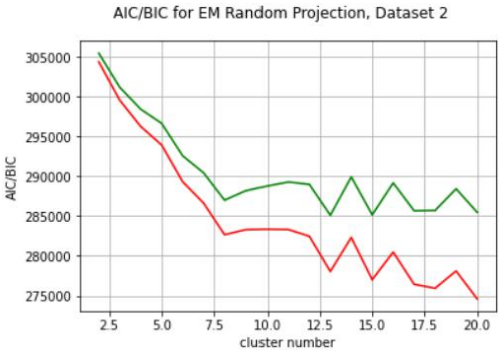








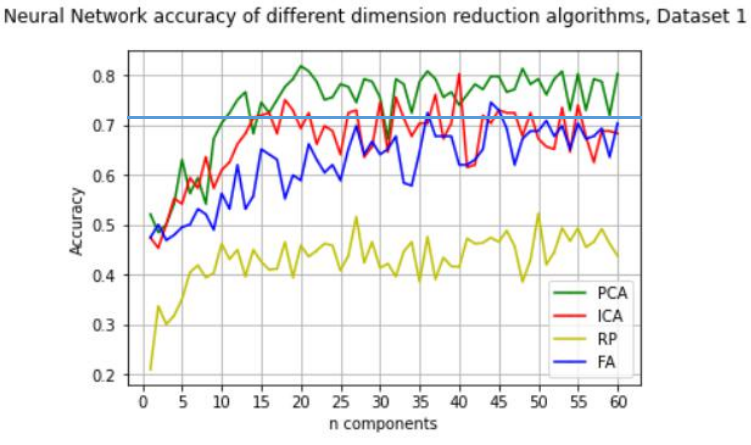


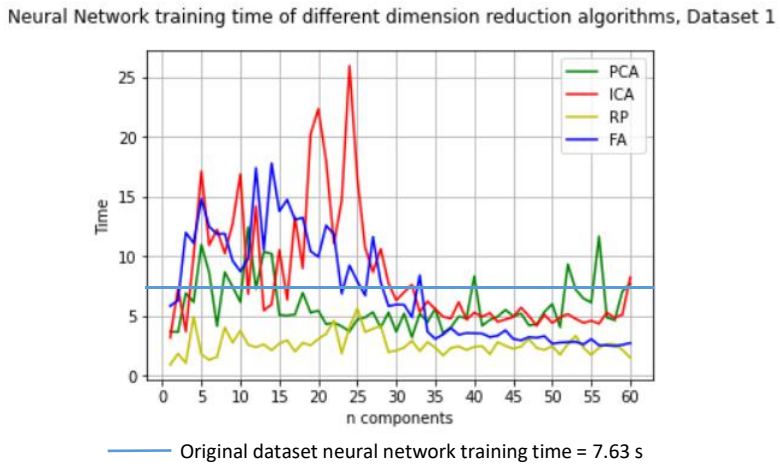


Part 4. Neural network learner on dimensionality reduced dataset

In this part of neural network training, I used the dataset after 4 dimensionality reduction algorithms. This is similar to what I did in part 2, but I will compare the 4 algorithms together and will compare the learning result with the result from the original data.

From assignment 1, I got the best parameters of neural network for dataset 1, and I will use the parameter for all the tests here. There will be 4 hidden layers and each one has 200, 100, 50, 25 units, the input units and output units count will depend on the attributes and classes counts. L2 penalty parameter is set to 0.01. Then I ran the test for different n-components dimensionality reduction algorithms and use the projected n-feature data to perform neural network learning. I separated the dataset into training/test set and test the final accuracy by the unseen test data. To get comparisons, I also ran the original dataset 10 times to get the average accuracy and training time.

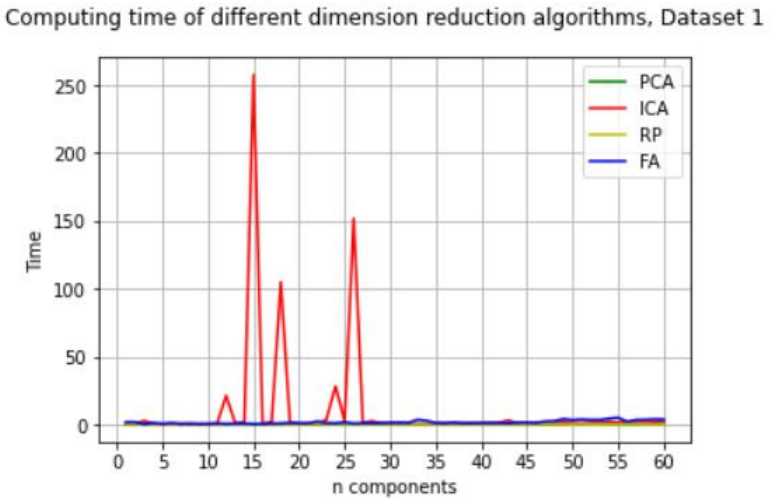


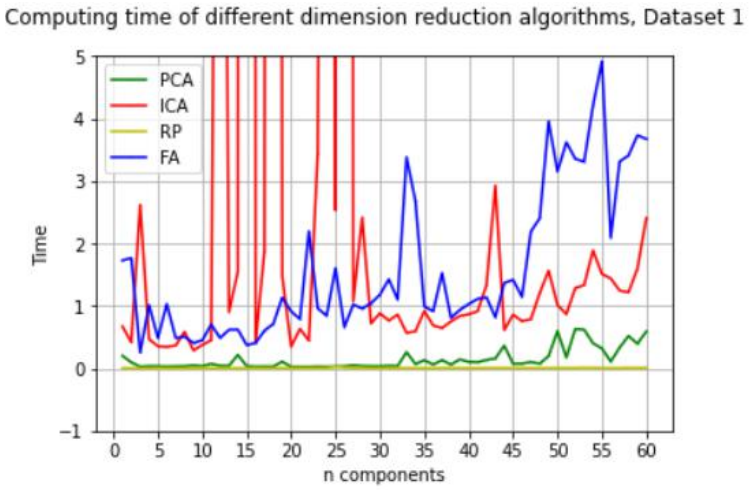


The original data's learning accuracy is 0.7163. we can tell from the chart the PCA has the highest accuracy score followed by ICA, then FA, the RP has the lowest accuracy. And once component counts exceed 20, the projected datasets can get a relatively high learning result, the dimension reduced a lot in comparison to the original 180-feature data. The PCA projected data is better than the original dataset, and ICA sometime performs better than the original dataset. The randomized project has the lowest accuracy, which may not be a good choice for this dataset.

For training time, the right chart shows the different algorithms projected data's training time, the original dataset takes 7.63 s to train. We can tell from the chart that RP projected data take the least time to train, and ICA projected data take the most time to train, this may be because of the independent features of the data. ICA, FA, and PCA curves are interleaved in the chart, but most of the time PCA projected data take less time to train compared with the others.

Overall, PCA performed the best in projecting data, the data achieve a good learning rate and take less time. The randomized projection projected data didn't get a high learning rate, even though it's fast in computing time.

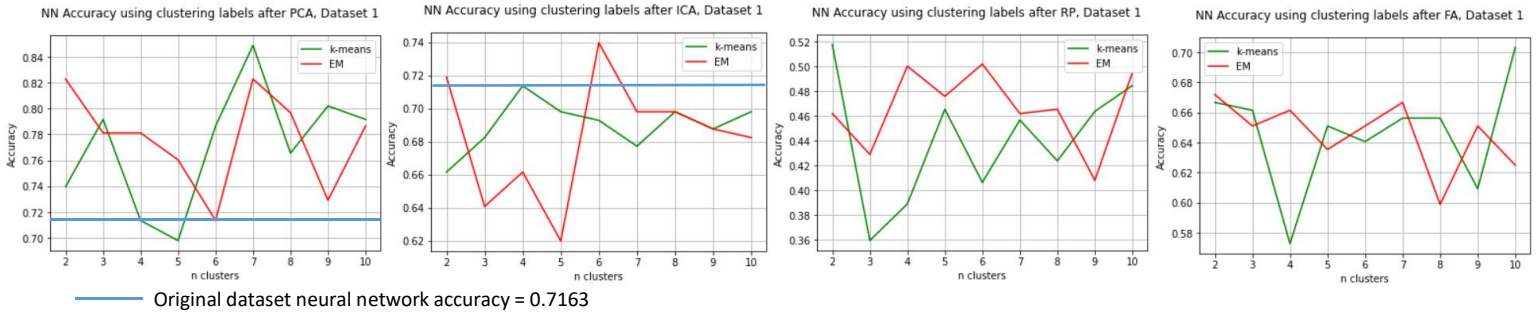




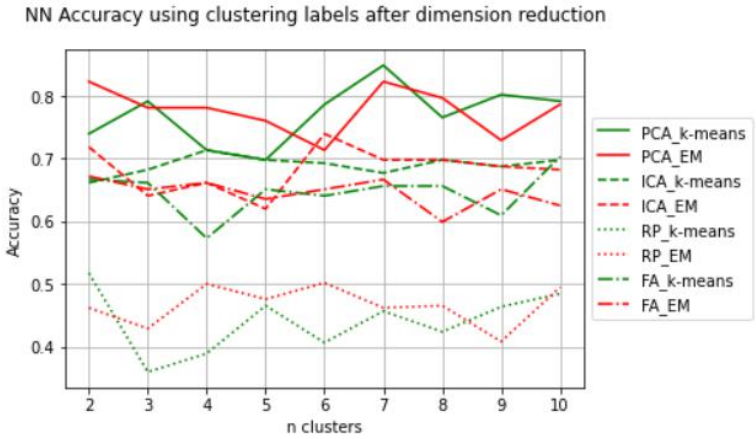
I also made a chart for the computing time of each dimensionality reduction algorithm, it calculated how long it take to transform the data. From the chart, the ICA takes the longest time, and sometimes it's hard to converge(for example, it didn't converge on n = 15). The random projection takes the least time to transform the data, PCA is very fast too. Overall, the computing time is ICA>FA>PCA>RP.

Part 5. Neural network learner on dimensionality clustered dataset

In this part, I first used the 4 dimensionality reduction algorithm to process the data, then use the 2 clustering algorithms(K-Means and EM) to cluster the data, and then add the cluster as a new feature and ran neural network learner on it. Based on part 4, when components count is 45 the dimensionality reduction algorithms all got a good neural network learning result compared to themselves. So I set the component count N = 45 in this analysis. The charts below show the learning rate of dataset, which included kmeans and EM clustering labels as a new feature, under different dimensionality reduction algorithms.



For the dataset projected by PCA, the peak for kmeans is the highest, it reached 0.8490 accuracy, higher than the original dataset 0.7163, also higher than the PCA projected data(without cluster label) 0.820. With EM label the accuracy(0.823) is also higher than without label. For ICA and FA, the data add clustering label is also higher than the original dataset. For RP, even though the learning rate is higher with the label, it didn't do better than the original dataset. The clustering label did help to improve the neural network learning rate.



The chart on the left shows the curves compared together, cluster numbers from 2 to 9. Overall we can see with the clustering label, the learning rate is PCA > ICA > FA > RP, and the PCA with kmeans or EM clustering label clearly did better than the original dataset. Instead, RP didn't improve the learning rate.

Conclusion

In this analysis, I explored the clustering algorithms kmeans and EM and tried different evaluation metrics. I also used dimensionality reduction algorithms to process 2 datasets, and used neural network learner on the transformed data, the result shows some transformed data obtained better results than the original data from assignment 1. In the last, I add EM/kmeans clustering label to be a new feature, and it helps to get better performance because it provides more information about the similarity among data points.

Compare all dimensionality reduction algorithms together, the PCA generally does the best, the ICA and FA are competing under different situations, these 3 algorithms reduce the dimensions and achieve a higher learning rate in neural network. RP is not working very well in the two datasets I chose, and it didn’ t improve the learning rate. Also, after the dimension reduction, the dataset is more likely to be clustered, EM generally works better than Kmeans in my datasets, and both of them can provide information to better the learning rate.