CS4641 Assignment3 Name: Shengrui Lyu GTID#: 903392423

Dataset:

Pima Dataset:

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. All patients here are females at least 21 years old of Pima Indian heritage.

These data frames contain the following columns:

npreg: number of pregnancies.

Glu: plasma glucose concentration in an oral glucose tolerance test.

Bp: diastolic blood pressure (mm Hg).

Skin: triceps skin fold thickness (mm).

Bmi: body mass index (weight in kg/ (height in m)\^2).

Ped: diabetes pedigree function.

Age: age in years.

Type: Yes or No, for diabetic according to WHO criteria. Which is our classification label.

Why it is interesting? This dataset is a build-in dataset in R language and it is wildly used in unsupervised learning area. And it has a bunch of features, and some of the features seem to be irrelevant to our target label based human intuition, which give me the chance to apply feature selection and transformation to see the result. Also, it is a binary classification problem,

Wine Dataset:

This dataset is from my assignment #1. The goal is to predict wine quality which has scores from (0 -10).

The attributes(features) are:

1 - fixed acidity: most acids involved with wine or fixed or nonvolatile (do not evaporate readily)

2 - volatile acidity: the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste

3 - citric acid: found in small quantities, citric acid can add 'freshness' and flavor to wines

4 - residual sugar: the amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweet

5 - chlorides: the amount of salt in the wine

6 - free sulfur dioxide: the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine

7 - total sulfur dioxide: amount of free and bound forms of S02; in low concentrations, SO2 is mostly undetectable in wine, but at free SO2 concentrations over 50 ppm, SO2 becomes evident in the nose and taste of wine

8 - density: the density of water is close to that of water depending on the percent alcohol and sugar content

9 - pH: describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic); most wines are between 3-4 on the pH scale

10 - sulphates: a wine additive which can contribute to sulfur dioxide gas (S02) levels, wich acts as an antimicrobial and antioxidant

11 - alcohol: the percent alcohol content of the wineee sulfur dioxide", "total sulfur dioxide", "density", "pH", "sulphates", "alcohol", "quality".

Why is it interesting? It is interesting because we can set the number of cluster up to 10. When we set cluster number to 10, the clustering problem may turn to a "classification" problem, depends on the class distribution within the clusters. Also, the dataset has 11 features, which is convenient for us to run feature selection.

Procedure 1: Clustering:

Pima:

I choose k = 2 at the beginning for this problem, because the label in my dataset is binary. And I use Euclidean distance for the distance function in each algorithm.

Result and Analysis for Simple **K-means**:

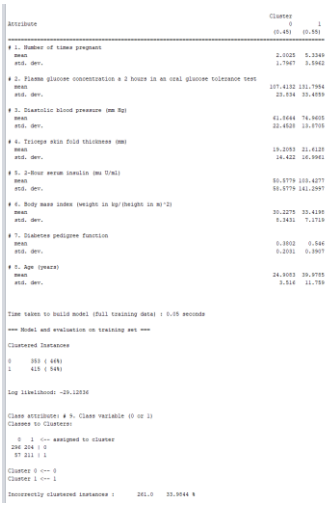
```
Final cluster centroids:
                                                                Cluster#
Attribute                                               Full Data           0           1
                                                          (768.0)     (515.0)     (253.0)
=============================================================================================
# 1. Number of times pregnant                              3.8451      2.0835      7.4308
# 2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test   120.8945   115.3282   132.2253
# 3. Diastolic blood pressure (mm Hg)                     69.1055     65.9903     75.4466
# 4. Triceps skin fold thickness (mm)                     20.5365     21.0194     17.9249
# 5. 2-Hour serum insulin (mu U/ml)                       79.7995     68.0194     69.1739
# 6. Body mass index (weight in kg/(height in m)^2)       31.9926     31.7751     32.4352
# 7. Diabetes pedigree function                            0.4719      0.4708      0.4741
# 5. Age (years)                                          33.2409     26.7725     46.4071


Time taken to build model (full training data) : 0.01 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      515 ( 67%)
1      253 ( 33%)


Class attribute: # 9. Class variable (0 or 1)
Classes to Clusters:

   0   1  <-- assigned to cluster
 380 120 | 0
 135 133 | 1

Cluster 0 <-- 0
Cluster 1 <-- 1

Incorrectly clustered instances :       255.0    33.2031 %
```

There are 380 items with label 0 (negative) and 135 items with label 1 (positive) in cluster #0, and 120 items with label 0 (negative) and 133 items with label 1 (positive) in cluster #1. The clustering algorithm did a good job in identifying negative labels, but didn't do well in identifying positive labels, since positive items are almost half distributed in cluster 0 and half distributed in cluster1.

To improve the clustering result, I tried to increase k.

After increasing k, the performance didn't increase, there is still not a cluster that contains mainly positive items. I think it is mainly due to the problem. Each feature doesn't differ that much for different labels, maybe I need to do the feature transformation to improve the performance.

Result and analysis for **EM**:



EM got 296 negative instances and 57 positive instances in cluster 0, 204 negative instances and 211 positive instances in cluster 1.

Compare with K-means, EM did a better job in identifying positive instances, and did a worse job in identifying negative instances, which is quite interesting. I think it is because of EM use probability to decide whether the instance is belong to cluster 0 or 1, and there are many instances with almost 50% probability of belonging to each cluster, and in K-means, these instances were clustered by random, and in EM, they were clustered strictly by the probability.

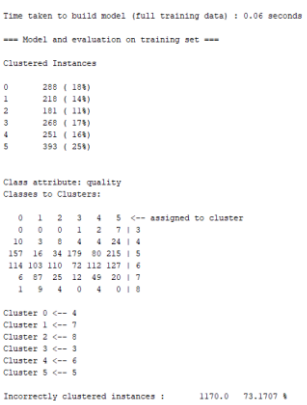
To improve the result, I think maybe we can change to another distance function which will fit to this specific problem, to avoid such many ambiguous instances.

Wine:

There are 6 values for quality label, which are 3, 4, 5, 6, 7, 8. However, 5 and 6 are the major two classes. Therefore, we can choose k as 2 or 6. And I will compare the performance below.

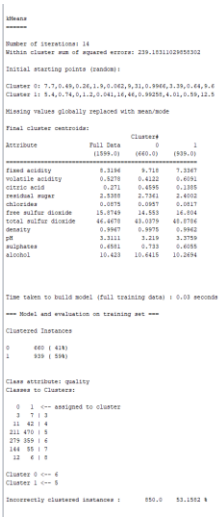
Result and analysis for K-means:

Choose K as 6:



We can see form the result, the clustering result is very bad. And incorrectly clustered instances are of 73% of total instances. I think it is due to the choice of k.

To improve the performance, I changed k to 2.

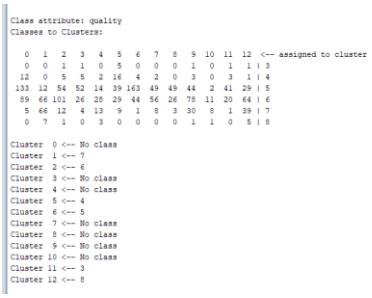


```
kMeans
======

Number of iterations: 14
Within cluster sum of squared errors: 239.1531102955302

Initial starting points (random):

Cluster 0: 7.7,0.49,0.26,1.9,0.062,9,31,0.9966,3.39,0.64,9.6
Cluster 1: 9.4,0.74,0,1.2,0.041,16,46,0.99258,4.01,0.59,12.5

Missing values globally replaced with mean/mode

Final cluster centroids:
                                    Cluster#
Attribute               Full Data           0           1
                         (1599.0)    (660.0)    (939.0)
=============================================================
fixed acidity             8.3196      8.716      7.9347
volatile acidity          0.5278      0.4122     0.6091
citric acid               0.271       0.4595     0.1305
residual sugar            2.5388      2.7361     2.4032
chlorides                 0.0875      0.0957     0.0817
free sulfur dioxide      15.8749     14.553     16.804
total sulfur dioxide     46.4678     40.0379    49.8706
density                   0.9967      0.9975     0.9962
pH                        3.3111      3.219      3.3759
sulphates                 0.6581      0.733      0.6055
alcohol                  10.423      10.6415    10.2694



Time taken to build model (full training data) : 0.03 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       660 ( 41%)
1       939 ( 59%)


Class attribute: quality
Classes to Clusters:

   0   1  <-- assigned to cluster
   3   7 | 3
  11  42 | 4
 211 470 | 5
 279 359 | 6
 144  55 | 7
  12   6 | 8

Cluster 0 <-- 6
Cluster 1 <-- 5

Incorrectly clustered instances :      850.0    53.1582 %
```

And the two clusters are dominated by class5 and class6. K-means did a better job on distinguish class5, however, it did poorly on distinguish cluster6. And it tends to put more class5 and class6 instances into cluster1, which is not good. Interestingly, it distinguish class7 well, most class7 instances are put into cluster0.

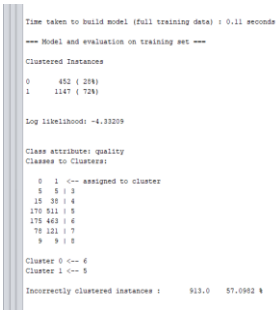
Result and analysis for EM:

When I didn't restrict number of clusters at first, the result is very poor:



```
Class attribute: quality
Classes to Clusters:

   0   1   2   3   4   5   6   7   8   9  10  11  12  <-- assigned to cluster
   0   0   1   1   0   5   0   0   0   1   0   1   1 | 3
  12   0   5   5   2  16   4   2   0   3   0   3   1 | 4
 133  12  54  52  14  39 163  49  49  44   2  41  29 | 5
  89  66 101  26  28  29  44  56  26  78  11  20  64 | 6
   5  66  12   4  13   9   1   8   3  30   8   1  39 | 7
   0   7   1   0   3   0   0   0   0   1   1   0   5 | 8

Cluster  0 <-- No class
Cluster  1 <-- 7
Cluster  2 <-- 6
Cluster  3 <-- No class
Cluster  4 <-- No class
Cluster  5 <-- 4
Cluster  6 <-- 5
Cluster  7 <-- No class
Cluster  8 <-- No class
Cluster  9 <-- No class
Cluster 10 <-- No class
Cluster 11 <-- 3
Cluster 12 <-- 8
```

And after that, I restrict the # of clusters to 2:



```
Time taken to build model (full training data) : 0.11 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       452 ( 28%)
1      1147 ( 72%)


Log likelihood: -4.33209


Class attribute: quality
Classes to Clusters:

   0   1  <-- assigned to cluster
   5   5 | 3
  15  39 | 4
 170 511 | 5
 175 463 | 6
  76 121 | 7
   9   9 | 8

Cluster 0 <-- 6
Cluster 1 <-- 5

Incorrectly clustered instances :      913.0    57.0982 %
```

The incorrect clustered instances decrease by 20 percent, however, the result is still not

good. Although it distinguishes class5 and class6 from other classes, it tends to put both in cluster1.

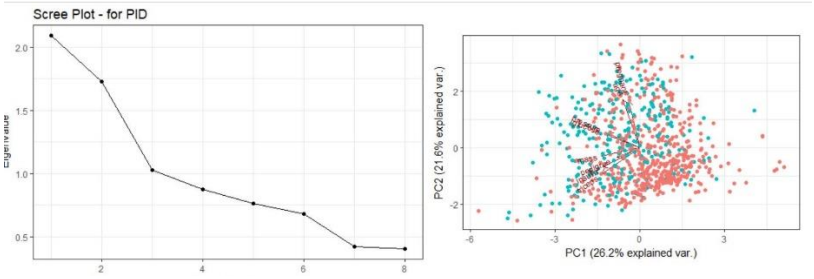
Comparison: Both K-means and EM didn't perform well in this problem, I think it is because our target label has too many values.

Procedure 2: Dimensionality Reduction

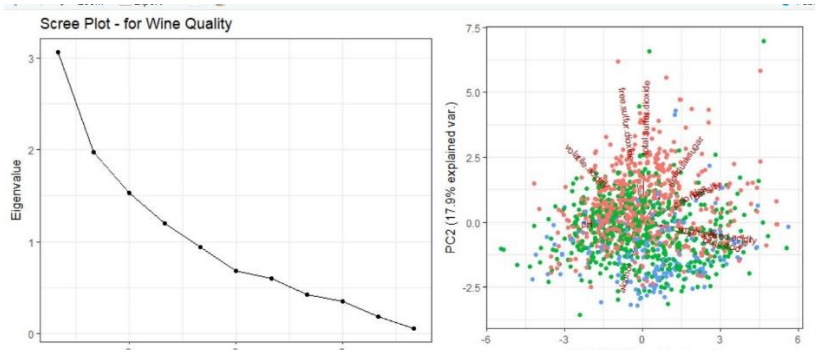
I use four algorithms for feature selection: PCA, ICA, Random Projection and Random Forest (Use the attributes that select by random forest).

PCA, eigenvalue distribution and visualization of data in first two component space:

Pima:



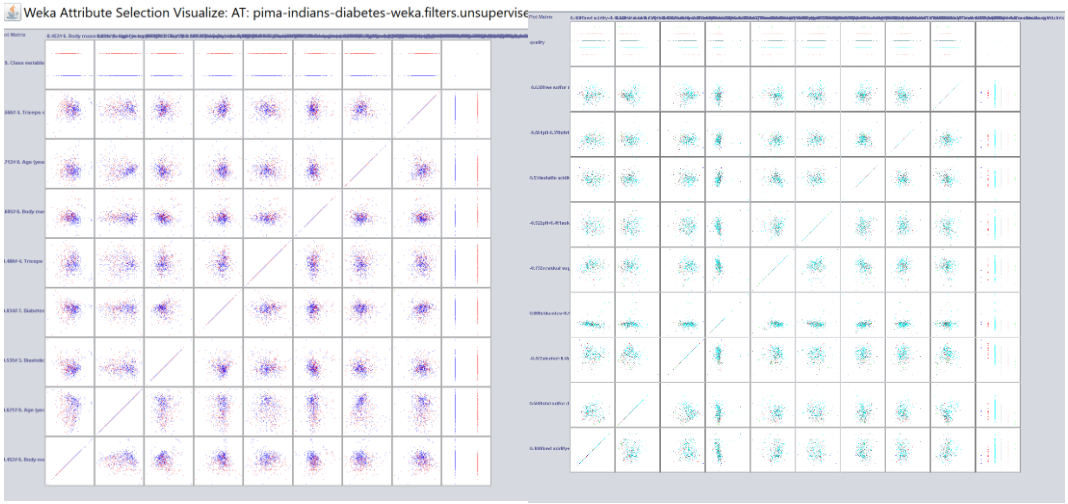
Wine:



The eigenvalue from the first principle component to the last component is monotonically non-increasing in both dataset. That is one of the PCA's property. We can also see that the pima dataset is separate better accordioning to the class labels than the wine data, which can be used to account for their performance difference in clustering after feature selection.

ICA, and visualization of data in the space

Pima: Wine:



The above two are the visualization of the data distributed in the transformed space. Since we have many independent components and we can only represent two as two labels in one plot, I use the matrix plot to visualize it.

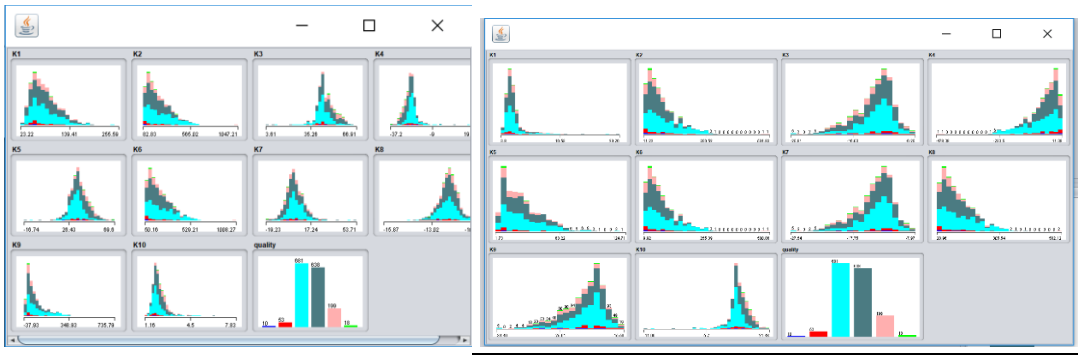
how kurtotic are the distributions: To analyze the kurtosis along each component, we need to look at the diagonal plots in the matrix plot. We can see from the diagram, the data is looks like a gaussian along the directions in Pima graph, but it is not the case for the Wine dataset.

Are the transformed space meaningful? I don't think they are meaningful, because I think for the two problem, there is no blind source behind the problems, at least the blind source is not as clear as the cocktail party problem that we studied in the lecture.

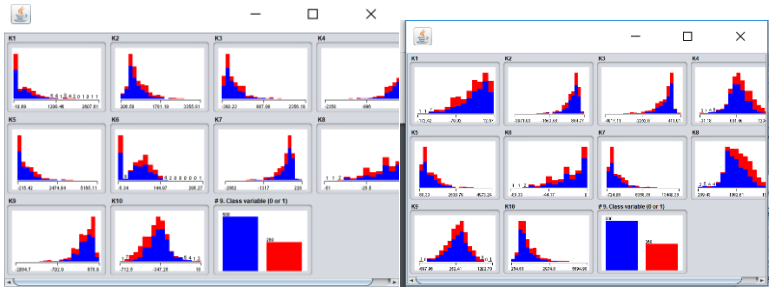
Random Projection:

Here are the visualizations for several iterations of Random projection.

Wine:

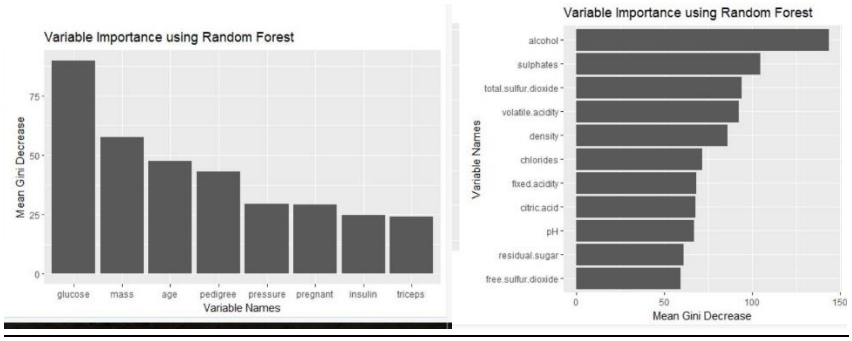


Pima:



From the visualizations, we can see that the variation between the two iteration of random projection is quite large.

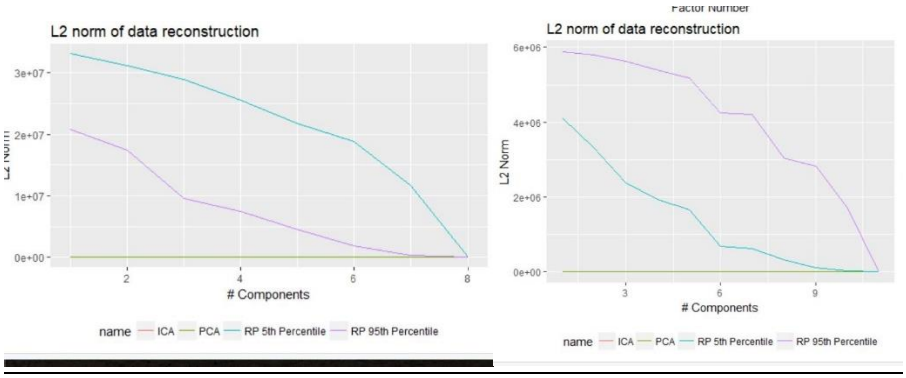
Random Forest Selection:



Pima is at the left and Wine is at the right.

We can see that the importance of the features that Random Forest algorithm "thinks" by looking at their Gini index. I think it is based on the information gain. In the Pima problem the most important feature is glucose, and in the wine dataset the most important feature is alcohol.

Data Reconstruction:



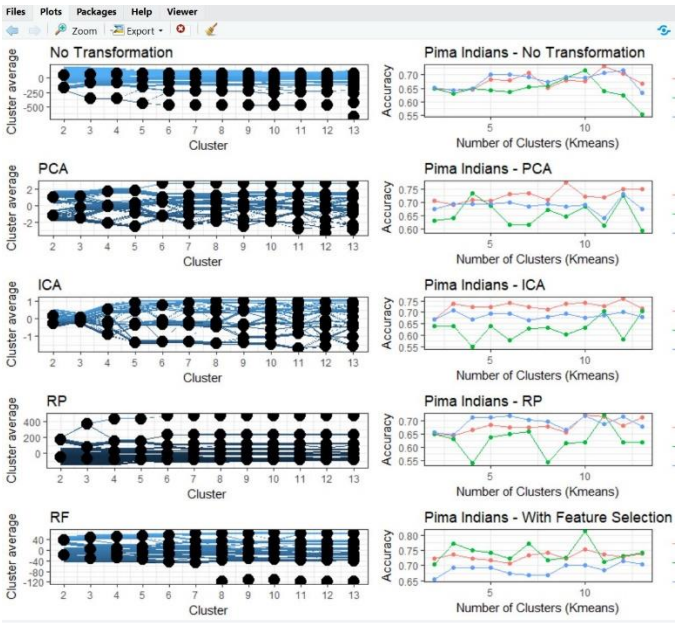
Pima is at the left and the Wine is at the right.

We can see that the curve of L2 norm which captures the reconstruction error in the above pictures. Obviously, the reconstruction error is decreasing when we use more components. And it is not surprising that the reconstruction error of PCA and ICA are ignorable compared to the error of RP. What's more, the reconstruction error of PCA should be the smallest because it is one of PCA's property that it will minimize the

reconstruction error.

Procedure 3: Reproduce Clustering Experiment:

Pima:



Red: Training Green: Validation Blue: Testing

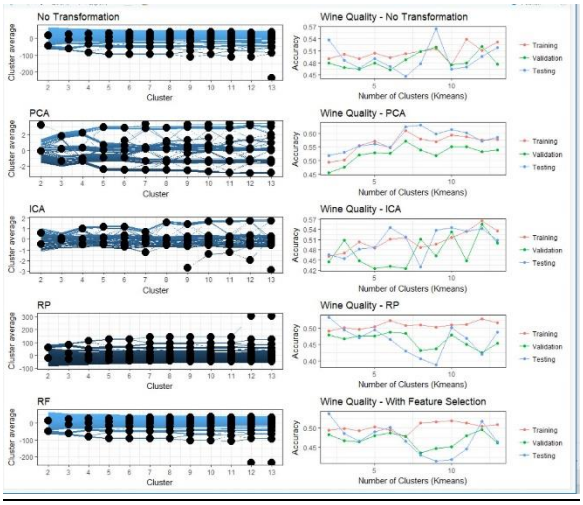
From the series of graph at the left side, we can see the effect of dimensionality reduction.

The graph is the clustering performance (accuracy) against the labels of the instances. And I show the accuracy curve with the k value as the x axis.

After the dimensionality reduction, we get different clusters. And the clustering with all four algorithms perform better than the raw clustering without any feature selection. And that's reasonable: Firstly, after dimensionality, we have different features and different values, therefore we will get different clusters after clustering. Secondly, the dimension reduction algorithms help us select or produce more relevant information, and it reduce the dimensionality, which will solve the problem bring by curse of dimensionality. Therefore, clustering algorithm like k-means will produce clusters that fit better with our label.

In average, the clustering after random forest gives us the best performance against the labels. That is because the feature selection process in random forest will take the label into account. Therefore, when we apply clustering on these features, it will give us a good accuracy against the label.

Wine:



The situation in wine dataset is not as desirable as the pima dataset. We can see that after random forest selection and random projection, the clustering performance is even lower. One reason is that this problem is much harder than the pima problem, because the output label is quality which is in range from 0 to 10. And the reason why the random forest performs poorly is not surprising, because when I applied random forest as decision tree in the assignment 1, it gave me a very low classification accuracy.

PCA gives us a slightly better performance than raw clustering, because the variance along original feature is very low, therefore it is hard to find a cluster boundary. However PCA will create feature that maximize the data variance along its direction, which I think will help clustering.

Procedure 4:

The wine dataset is the dataset that I use in my assignment1. So please see the above discussion for the performance of dimensionality reduction on this dataset.

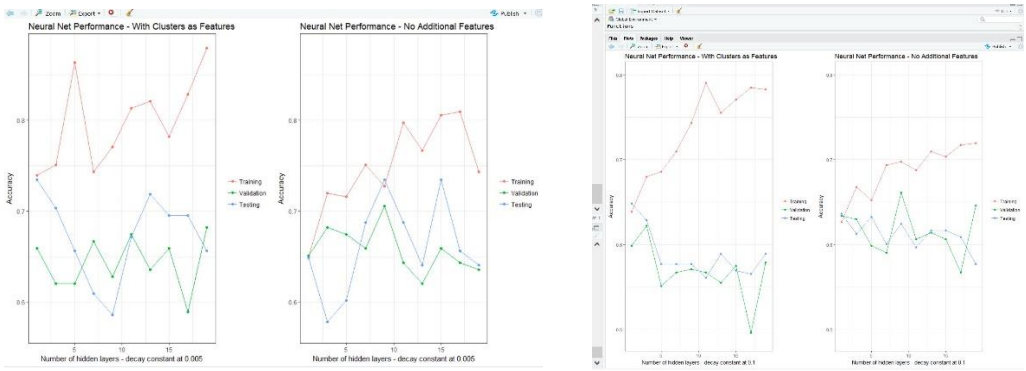
Re-run the neural networks problem:

	Training time	Training Accuracy	Testing Accuracy
Original	202	0.53	0.45
Re-run	161	0.61	0.64

The above table shows that after we re-run the neural networks with the newly projected data, all the aspect was improved a lot. That is because the newly projected data uses different features. And these features provided more information than the original features, and the total number of features is reduced, therefore the curse of dimensionality issue was solved and the training time was shorter.

Procedure 5: Neural Nets with Cluster as Feature

Pima: Wine:



The behavior of the two sets of curves is very similar.

From the graph we can see that the training accuracy is improved when we use cluster as a feature in neural networks. However, that is not that meaningful.

For testing accuracy, we can see that when hidden layers is low, the NN with cluster performs better, when number of hidden layer increases, the advantage of using cluster as feature disappears. I think maybe when the hidden layer increases, it will perform kind of same function as clustering, that's why the peak in NN-Clustering graph is as high as the peak in the NN-Original graph. And when the hidden layer increase in NN-Clustering, something like overfitting happens, therefore the testing accuracy decreases rapidly.

Therefore, I suggest using less hidden layer when apply clustering result as a feature and use more hidden layer when there is no clustering result available.

Problem	Training Time without Clustering	Training Time with Clustering
Pima	112.04	90.54
Wine	202	161

Also, from the above table, we can see that when we use clustering result as a feature, the training time decreases for the problems, because the clustering result for pima agrees with the label a lot, which provide a lot of information for classification.