Assignment 3

Vinodh Kumar Sunkara,
Microsoft Redmond, WA

As part of this assignment, we've two main splits:

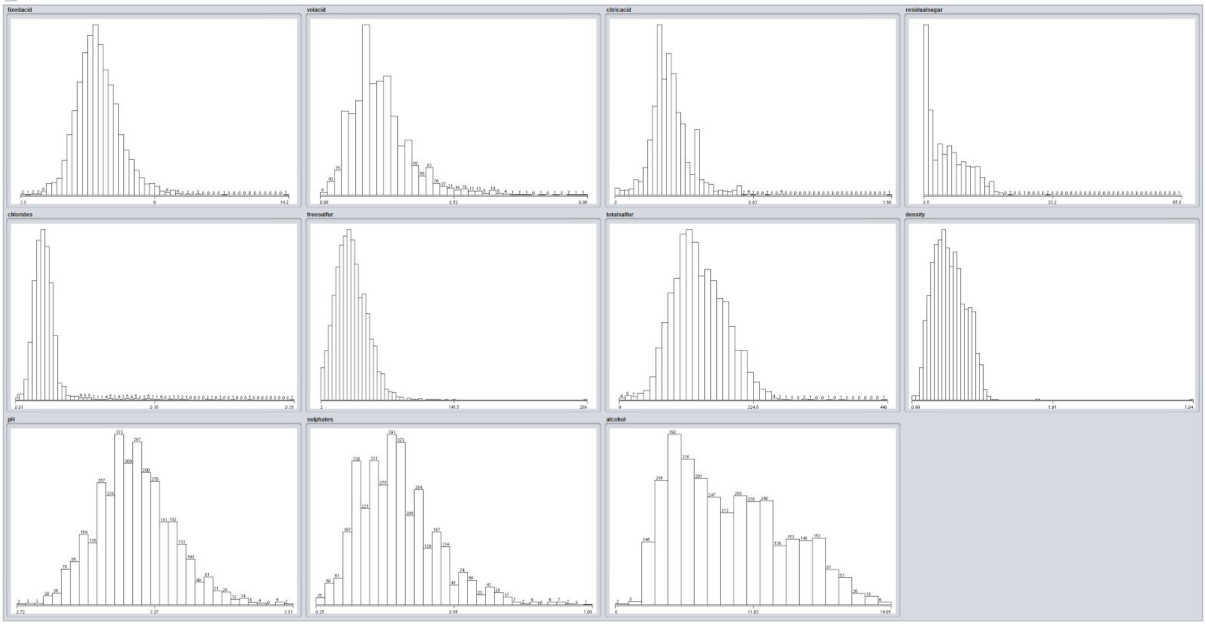
1. Running following clustering algorithms on two data sets
    a. K-Means clustering
    b. Expectation Maximization
2. Running following dimensionality reduction algorithms
    a. PCA
    b. RCA
    c. ICA
    d. Any other feature selection algorithm: ISCA (Insignificant component analysis)

I have used Weka for all my experiments. ICA wasn't available in Weka and I installed the Students.Filters weka plugin containing the FastICA algorithm implementation and that added missing IndependentComponents filters under unsupervised attribute filters.

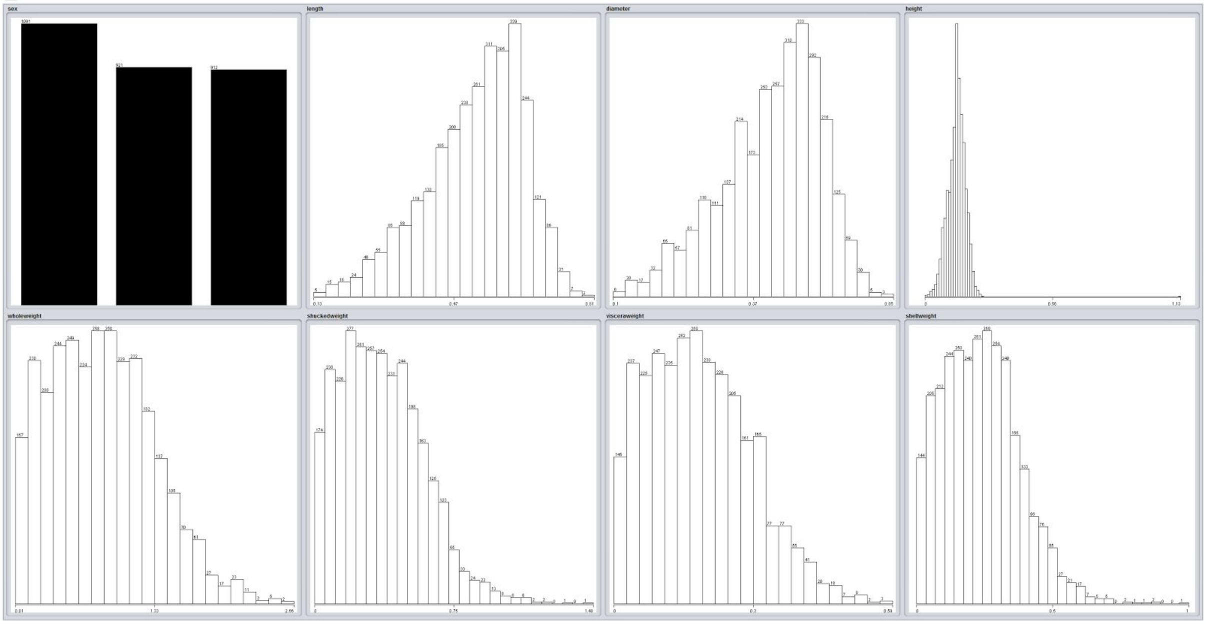
Datasets

I've considered Abalone and WineQualityWhite data sets. First data set is continuous and wine quality is mostly discrete. Both the data sets were used in first assignment and it made more sense to use same data sets here for better interpretation of results and data from experiments.

Wine Quality: This data set is related to red and white variants of the Portuguese "Vinho Verde" wine. The physiochemical (inputs) and sensory (output) variables are available as part of this dataset. The classes are ordered and not balanced – for eg, there are many normal wines other than excellent and poor ones. Outlier detection algorithms could be used to detect the few excellent and poor wines. It has 12 attributes in total and 4898 samples to compute across. Along with the practical advantages to detect excellent/poor wines, the dataset is interesting in terms of results obtained when unsupervised algorithms are applied. This is a challenging task to be handled by any learning algorithm over a broad range of output. This dataset is also not known to have very good results with any unsupervised learning model, thus making the dataset useful for critiquing learning techniques. Below is the pictorial representation of data attributes. Since, unsupervised learning under experimentation, we don't have the class attribute in the training data.



Abalone: It’s a dataset taken from a field survey of abalone, a shelled sea creature. The task is to predict the age of abalone given various physical statistics. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope -- a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age. Further information, such as weather patterns and location (hence food availability) may be required to solve the problem. Abalone have gone through a period of significant decline in recent years due to both natural and unnatural environmental pressures (reintroduction of sea otters, loss of habitat). If techniques to determine age could be used that did not require killing the abalone in order to sample the population, this would be beneficial. The abalone dataset is of interest in many ways because it is so resistant to good performance under machine learning techniques. Below is the data attributes sampled out.



Clustering Algorithms

We run K-Means and Expectation maximization clustering algorithms on Abalone and WineQuality DataSets.

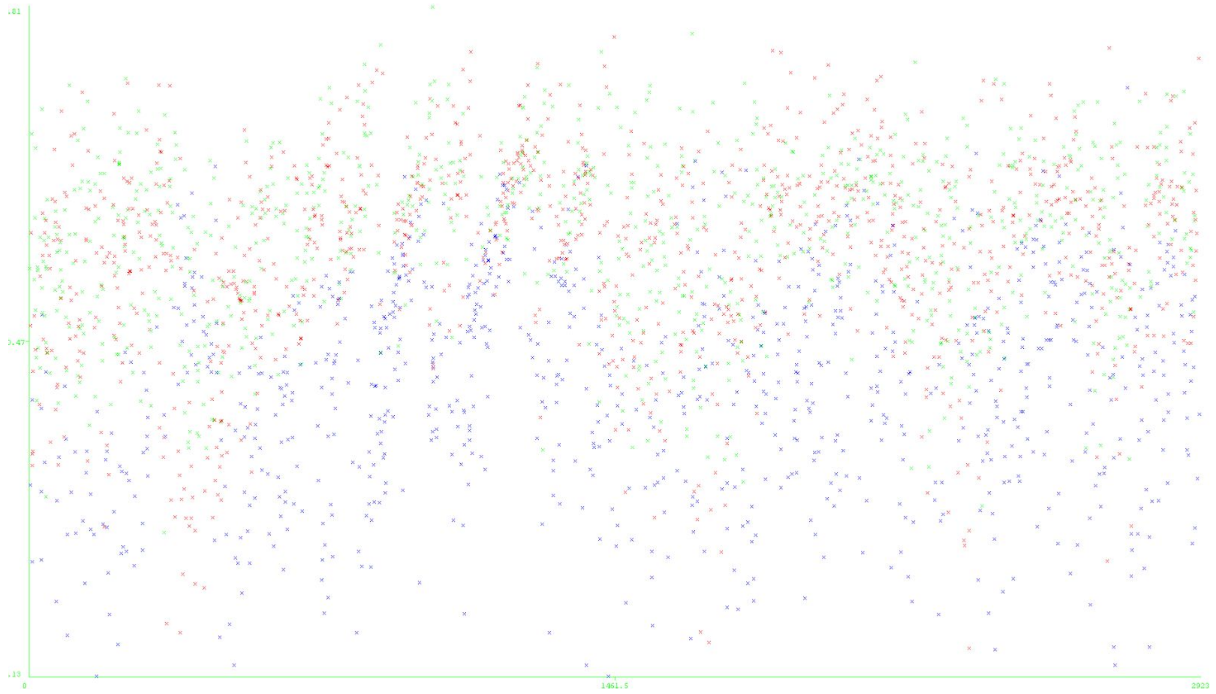
Parameters Selection

Cluster number: Since I want to clearly observe how our data set is split by the clustering algorithms, I would like to choose cluster number accordingly. Giving it too less number will make it hard to study the split and giving it too large will end up too less data in each cluster. Hence, I selected 3 clusters for Abalone and 2 clusters for Wine Quality.

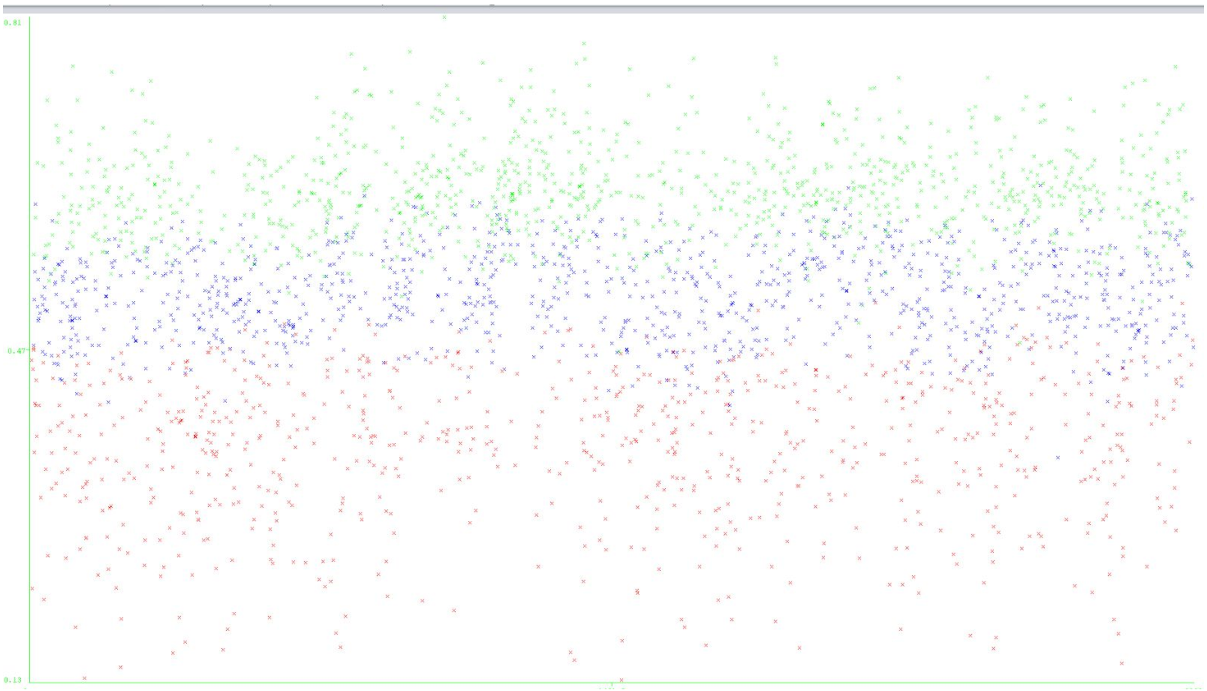
Distance Measure: We can use Hamming distance measure for data with missing features. Since our data doesn't have any missing values for features, I've chosen Euclidian distance measure.

Below are the scatter plots for both data sets using kmeans and EM clustering algorithms, run on weka tool. Instructions are mentioned in the README file. These plots are generated by first applying PCA and then keeping only first four eigen vectors and reducing the data sets to 2D

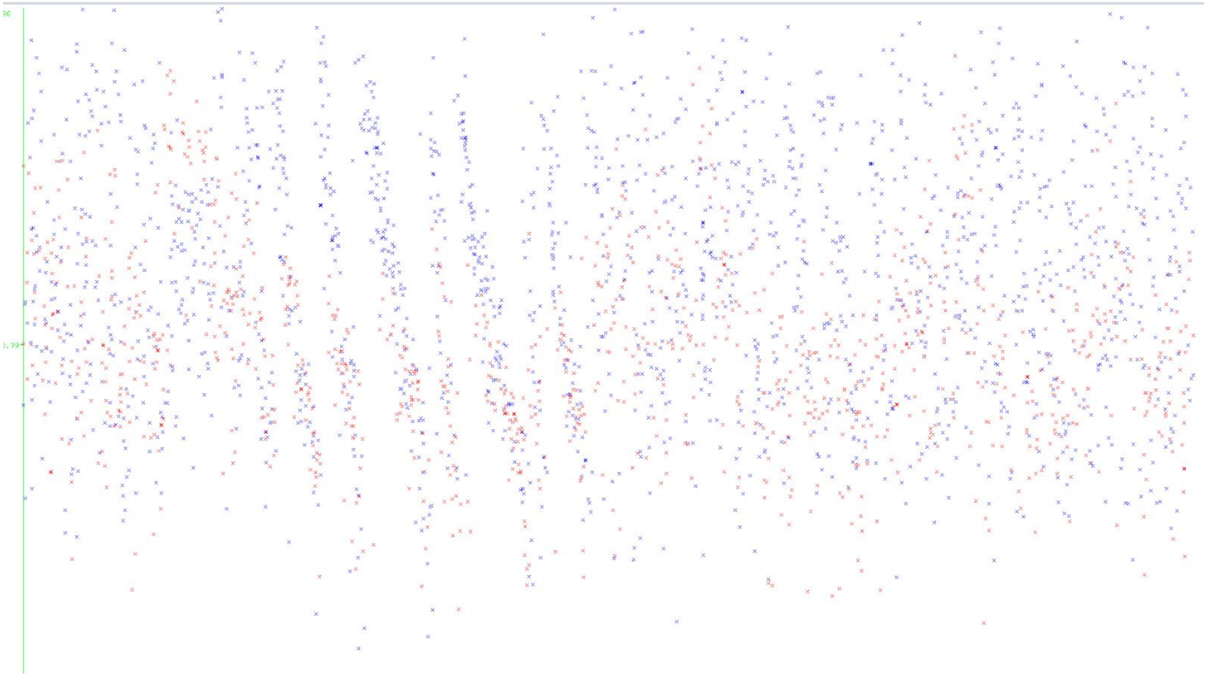
Abalone KMeans:



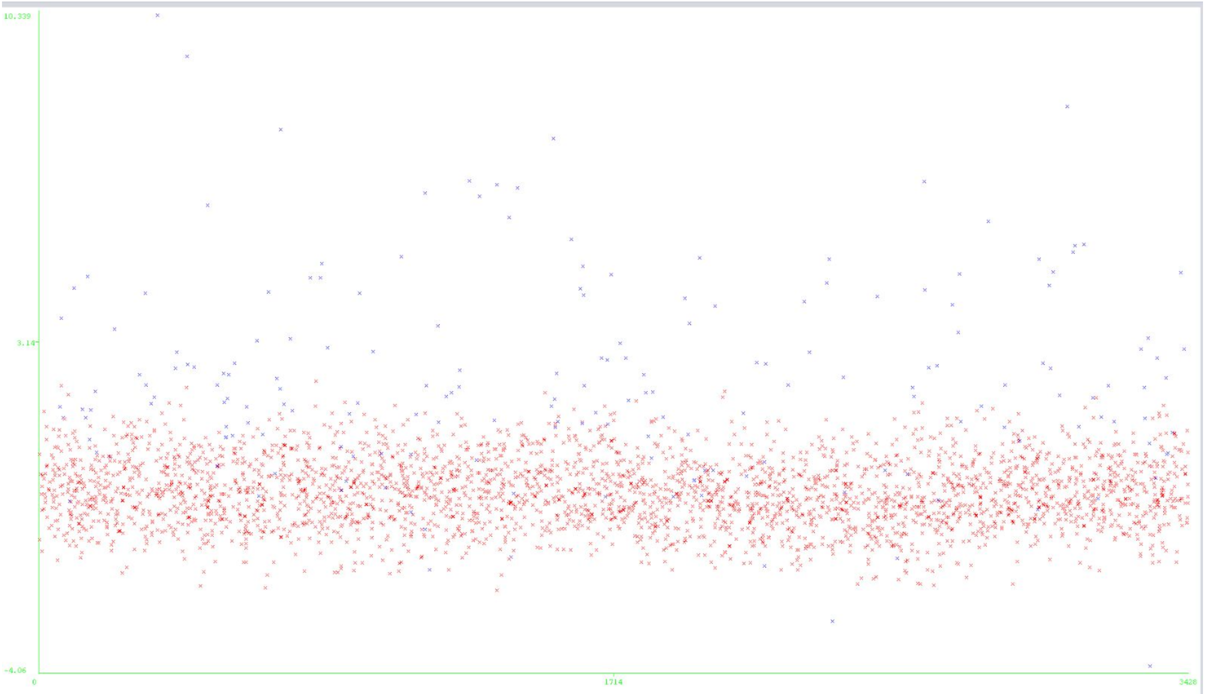
Abalone EM



WineQuality KMeans:



WineQuality EM:



We can see from above that the Expectation maximization maximizes the data point membership. EM runs initially a clustering algorithms to to generate cluster centers and then it does iterations to update expectations until it reached threshold. The result is clear from the graphs. In Abalone, K means mixed some green and red cluster points. We can see that issue is completely fixed in Abalone EM plot. Green and red cluster points are separated out completely. Similarly in WineQuality dataset, EM clustering results are improved from KMeans.

Evaluation:
Following few other metrics are used to evaluation the clustering. The formulas used for implementing these metrics have been taken from Evaluation of Clustering and Davies-Bouldin Index.

Dataset	Algorithm	Purity	NMI	DBIndex	Time(ms)
Abalone	KMeans	0.964259	0.031149	0.337489	49
Abalone	EM	0.964259	0.031253	0.294183	308
WineQuality	KMeans	0.694213	0.159276	0.591324	492
WineQuality	EM	0.694295	0.154210	0.338746	685

Davies-Bouldin Index is an internal metrics which is a ratio of cluster scatter and cluster separation and doesn’t use any labels. A lower DBIndex value means a good cluster. Using this metrics also, EM wins over KMeans.

External evaluation metrics use the labels to evaluate clustering. Purity is an external measure which is used to measure how much a majority class compared to other classes present in a cluster. It’s values lies between 0 and 1, and higher the purity values indicates better clustering. Abalone has good purity measure than wine quality.

When each instance has it’s own cluster, Purity measure is flawed. In such a case, NMI - Normalized Mutual Information measure can be used. This value is higher for wine quality data set than abalone.

Dimensionality Reduction

We selected the eigen values based on 85th percentile rule. We consider only those eigen values from top in descending order whose sum corresponds to 85% of total eigen values sum. We can also use some other rules like setting up a threshold of a percentage in top most eigen value and remove all other eigen values that are lesser than the threshold. Based on this rule, we get four dimensions for abalone and two dimensions for wine quality dataset. Lets see the PCA output:

```
PCA on WineQuality:
EigenValues: Vector(17.09243334032249, 16.469488022552785, 0.392771448058647,
0.237005695197132, 0.092771448058647, 0.002771448058647, 0.000771448058647,
0.0000631022356477, 0.0000031026251777, 0.0000001026054377)
Dimensions to keep: 2, after considering 85.0% variance from Eigen Values

PCA on Abalone:
EigenValues: Vector(21.7938338371087755, 20.22939083129543682,
20.89394707412974O387, 19.0029008320658867686, 0.001052356474803368,
4.885092337580203E-4, 4.2642099595839055E-4, 1.4775340983798694E-4)
Dimensions to keep: 4, after considering 85.0% variance from Eigen Values
```

RCA

I used RandomProjection filter from weka with nxm random matrix where n is number of components to generate and m is the number of original features. Total n vectors of size m are formed. All instances are projected on these n vectors. I’ve my n set to 2.

ICA

We’ve a parameter to set the number of independent components. Keep it as -1 will take total number of independent features as the components. I’ve set this value as 3.

ISCA

Insignificating component analysis is opposite of principal component analysis because it picks bottom few eigen values instead of top ones. I've picked bottom 4 based on a threshold value.

Divergence from initial dataset

After applying the dimensionality reduction algorithms, squared sum and RMSE gives total divergence from initial dataset. Squared sum and RMSE gives the total elements wise error between the initial dataset and the obtained dataset after applying the dimensionality reduction filter. Below are the squared sum and RMSE approximated to two decimals.

Dataset	Filter	SquaredSum	RMSE
Abalone	PCA	970.23	0.19
Abalone	ICA	3.59	8.29
Abalone	RCA	27932.83	0.92
Abalone	ISCA	5.4	1.54
WineQuality	PCA	78342.92	0.69
WineQuality	ICA	9.42	8.15
WineQuality	RCA	1920382.82	3.96
WineQuality	ISCA	2.92	3.02

Different cluster evaluation metrics for abalone and winequality datasets corresponding to all four dimensionality reduction algorithms using kmeans and EM clustering mechanisms are shown below:

DimRed Algo	Dataset	Clustering Algo	DBIndex	Purity	NMI	Time(ms)
PCA	Abalone	KMeans	0.06	0.95	0.034915	24
PCA	Abalone	EM	0.07	0.95	0.034992	530
RCA	Abalone	KMeans	0.07	0.95	0.034915	14
RCA	Abalone	EM	0.05	0.95	0.034982	42
ICA	Abalone	KMeans	1.68	0.95	0.034931	59
ICA	Abalone	EM	4.36	0.95	0.068192	1028

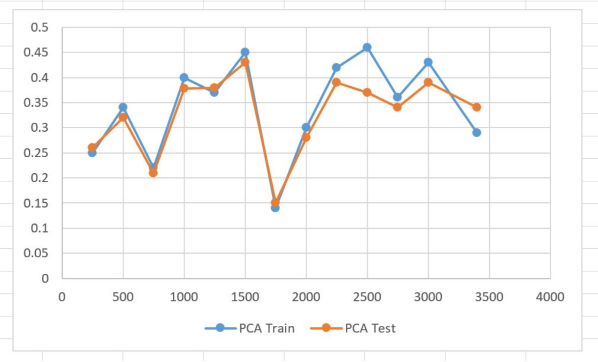
ISCA	Abalone	KMeans	0.37	0.95	0.034998	35
ISCA	Abalone	EM	0.38	0.95	0.034981	1035
PCA	WQ	KMeans	0.57	0.81	0.271303	194
PCA	WQ	EM	0.58	0.83	0.035194	812
RCA	WQ	KMeans	0.21	0.63	0.000412	49
RCA	WQ	EM	0.21	0.68	0.000382	93
ICA	WQ	KMeans	4.64	0.72	0.000612	58
ICA	WQ	EM	2.39	0.74	0.165239	294
ISCA	WQ	KMeans	0.81	0.89	0.254193	71
ISCA	WQ	EM	0.37	0.81	0.268772	250

RCA and ICA degraded the cluster qualities but RCA does it bit lesser than ICA. There isn’t drastic change in different metrics in case of PCA before and after dimensionality reducation algorithm. PCA is the best choice for reducing the dimensions as per my above analysis.

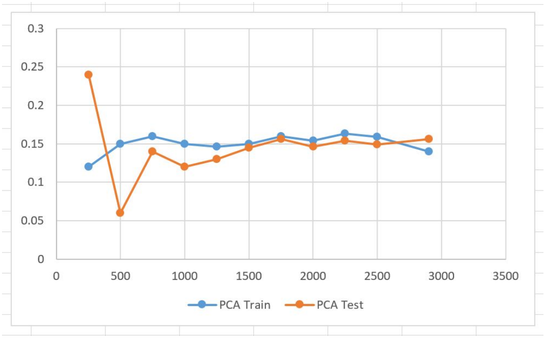
Neural Networks

After dimensionality reduction algorithm applied on both data sets, we can have the neural networks error graphs below. We had already seen abalone dataset with around 30% error rate earlier and with the application of dimensionality reduction, error rate is converged to around 19%. Of the four dimensionality reduction algorithms, PCA gives best performance. Below are the different error rates for neural networks after applying different dimensionality reduction algorithms on both data sets.

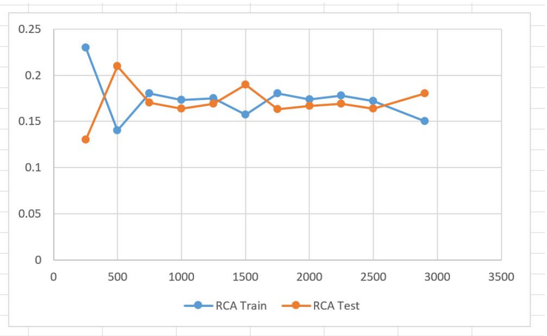
Abalone PCA



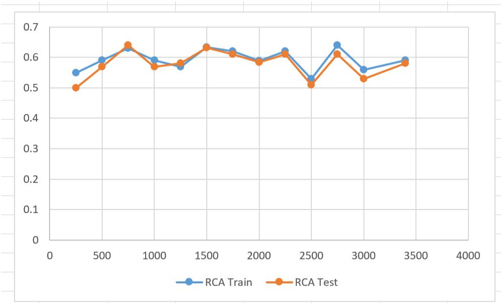
Wine Quality PCA



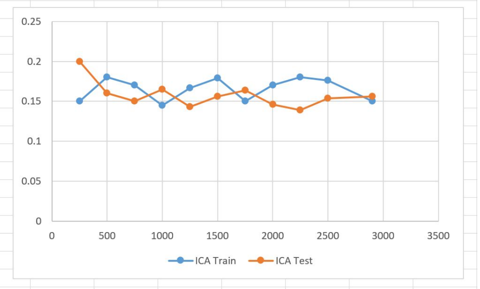
Abalone RCA



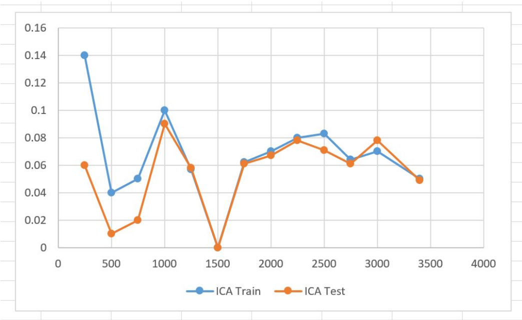
WineQuality RCA



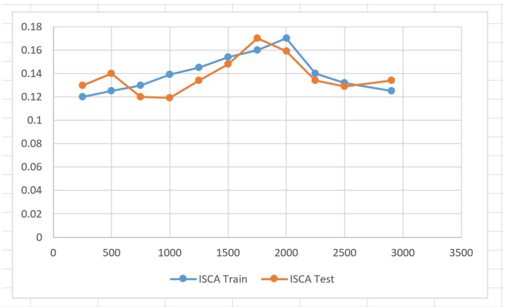
Abalone ICA



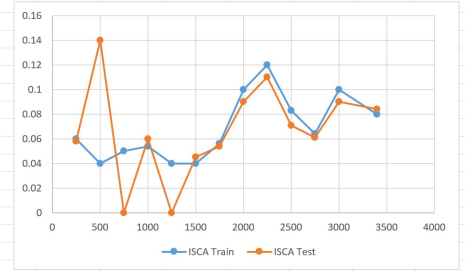
WineQuality ICA



Abalone ISCA



WineQuality ISCA



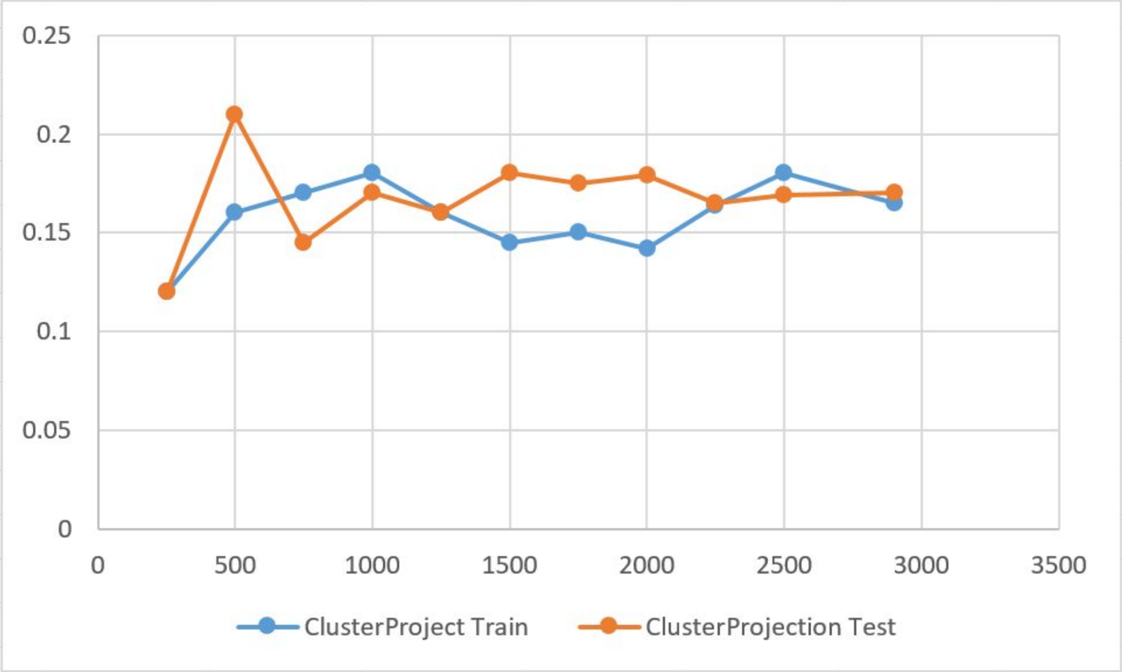
Cluster output as dimensionality reduction

I've taken both the cluster outputs as two new features and initial label as the label. So, the corresponding dataset converts into:

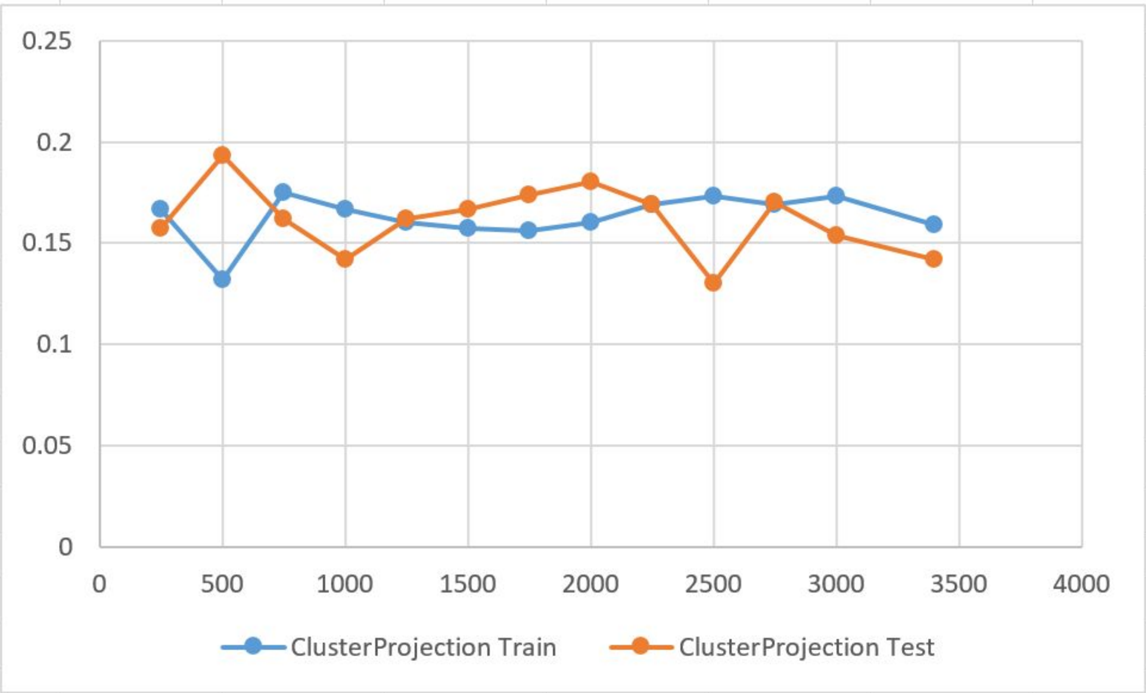
Feature1: cluster id of kmeans
Feature2: cluster id of EM
Label: initial label

Now neural network is run with backpropogation and the accuracy is same as what we had with dimensionality reduction.

Abalone Cluster Projections



Wine Quality Cluster projection



Conclusion:

Dimensionality Reduction has produced a good summarization over data with better accuracy than original datasets. Clustering output also worked well in performing dimensionality reduction without compromising on accuracy part.