Jassimran Somal (jkaur9)

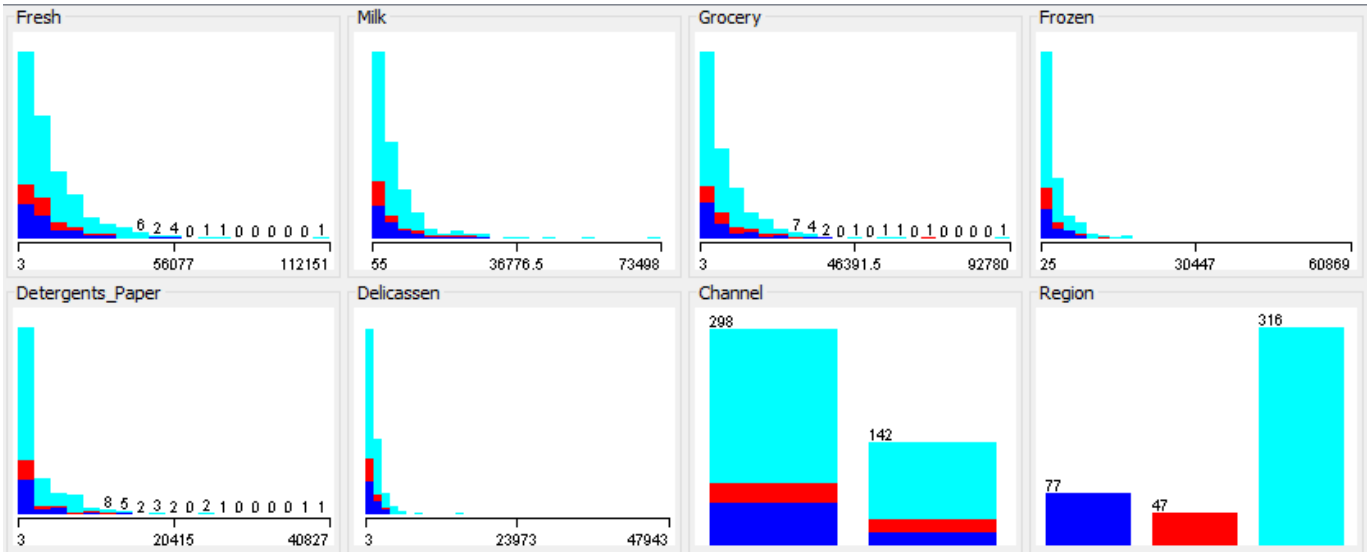
Machine Learning - Assignment 3

DATASETS

Whole Sale Customer Data Set: This dataset is obtained from UCI dataset repository and it has 8 attributes and 440 instances. It is a classification problem where Customer's region can be classified as 1,2 3 based on their purchase of goods. This is the new dataset I chose because all attributes in my assignment 1 dataset were categorical and clustering with proper distance measures would have been challenging.

Why it's interesting?

Having domain knowledge is really important to not only excel but one needs to know what you are really looking for. Hence, customer dataset was very simple to understand. This dataset has all numeric values which will make it more effective for running clustering problems using Euclidean distance as a measure of grouping data. Also, some dimensionality reduction algorithms like Independent component analysis will yield much more meaningful results with numeric data rather than categorical values. Although 8 attributes are not a lot but it will be interesting to see if dimensionality reduction algorithms will help in any way by minimizing the curse of dimensionality. Dataset Distribution is shown below:



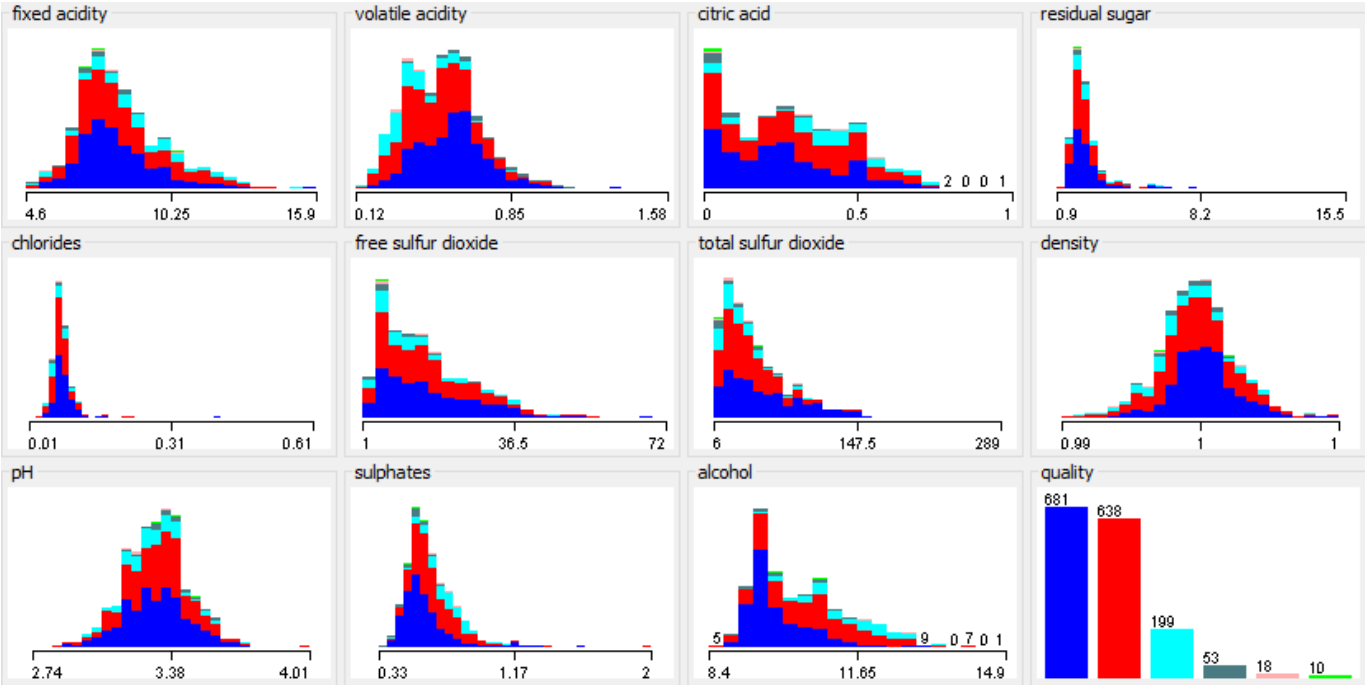
Red Wine Data Set: This dataset is obtained from UCI dataset repository and it has 12 attributes and 1599 instances. The task is to classify the wine quality as a score between 0-10 based on physicochemical and sensory variables but current dataset has score values from 3-8 only. This dataset is from my assignment 1.

Why it's interesting?

Firstly, it gives me a chance to understand wines better and what make them all so different. Also, classes are not

balanced as there are much more normal wines than excellent or poor one. And it has to be classified over a range of 6 different values, so giving me chance to see how algorithms behave differently for 6 classifier values.

This dataset is interesting because since my first assignment and second assignment, this dataset has been very resistant and I never got performance accuracy better than 64%. It would be interesting to see if clustering or dimensionality reduction algorithms can help improve accuracy in any way. M hoping to understand if the reason for poor performance all this while was curse of dimensionality.



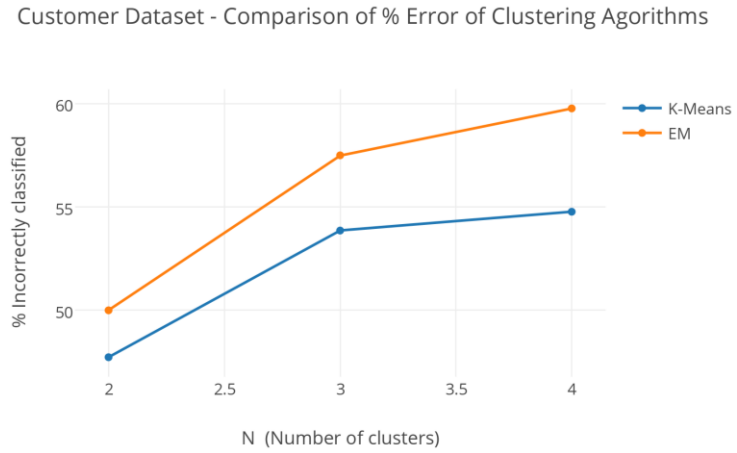
Clustering Algorithms:

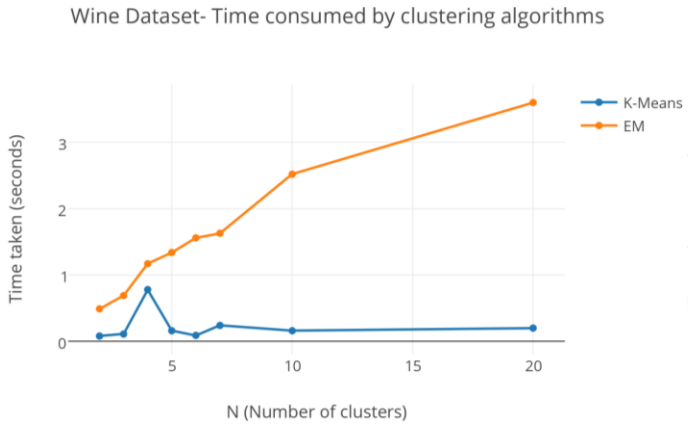
K-Means Clustering and Expectation Maximization:

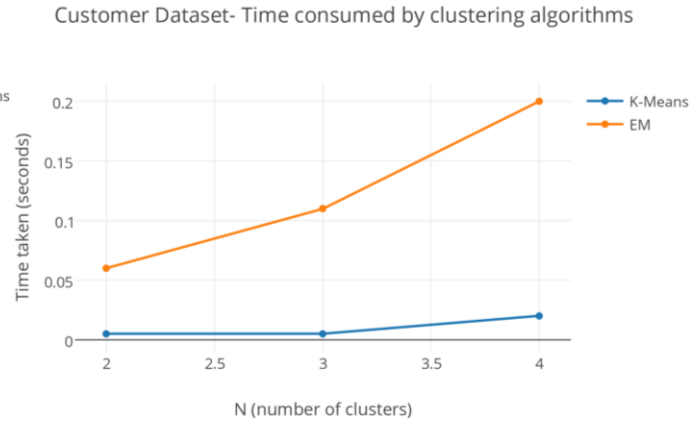
Since my datasets are classification problems, in order to choose k, I used classes to clustering evaluation and noticed how well the new clusters align with original labels. So, I experimented with different number of clusters and each time noticed the % of labels incorrectly classified. Below experiments show that the number of incorrectly classified classes increase as the number of clusters increase. Wine Dataset has 6 different classes and shows approximately 20% more error when clustered into 6 clusters versus 2 clusters. This could be happening because most of the wines fall into two classes (quality value - 5 & quality value - 6) and more than 82% samples fall into either if the two categories. Due to poor distribution of classes, they align better into two clusters. Customer dataset has 3 different classes and but it still shows better accuracy with 2 clusters than 3 clusters. Again it could be attributed to uneven distribution of classes as one class (region value - 3) has more than 75% samples.

K-Means clearly gives better results in Customer Dataset, while in Wine Dataset, error is higher with Expectation Maximization with smaller number of clusters but it starts performing better as number of clusters increase. It should further be noticed that for both the datasets, K-Means clearly outperformed Expectation Maximization in

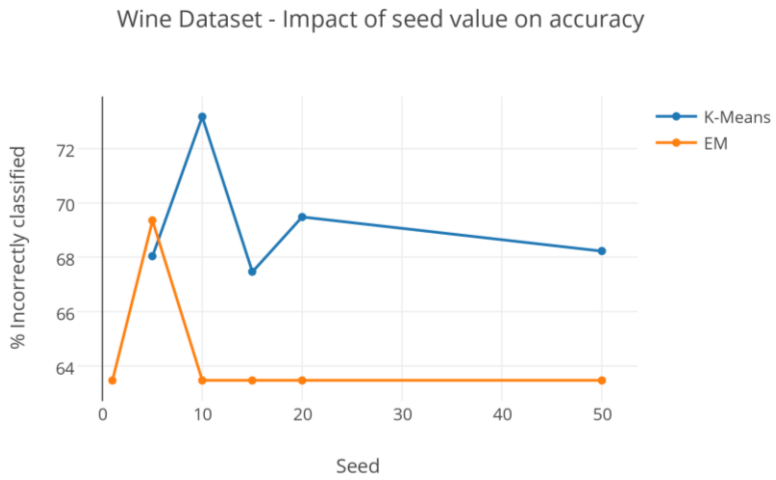
terms of training times for clustering. I choose to go forward with K = 2 for my experiments as it gives most accuracy for aligning up with classes for classification evaluation and it is yields results faster.



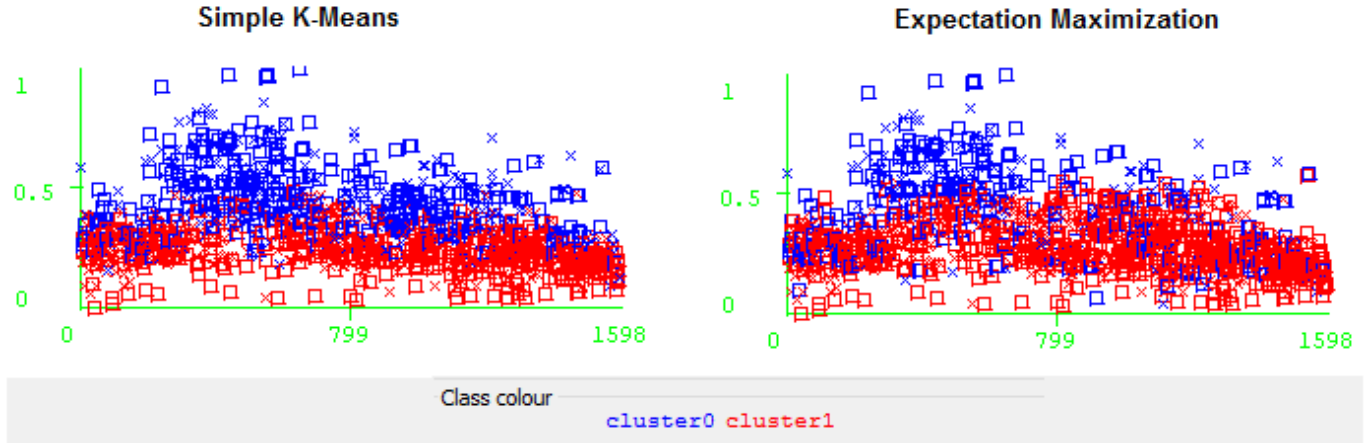




The sum of squared errors within the clusters decrease with increase in number of clusters. Increasing the number of seeds in Expectation maximization has little to no impact on the accuracy but it a right choice of seed does improve accuracy in K-Means. Hence I choose seed = 15 for K-Means and 10 for Expectation Maximization



For the purposes of this analysis and to maintain consistency for the experiments, I used Euclidean distance measure as a similarity measure between instances because Manhattan distance gave slightly worse results.



For Wine Dataset, For two clusters Simple K-Means performed better than Expectation Maximization by 6%. But clusters are overlapping showing overall poor clustering performance.

For customer Dataset, K-Means performed better by 7% than Expectation Maximization but over-all the clustering is still very overlapping.

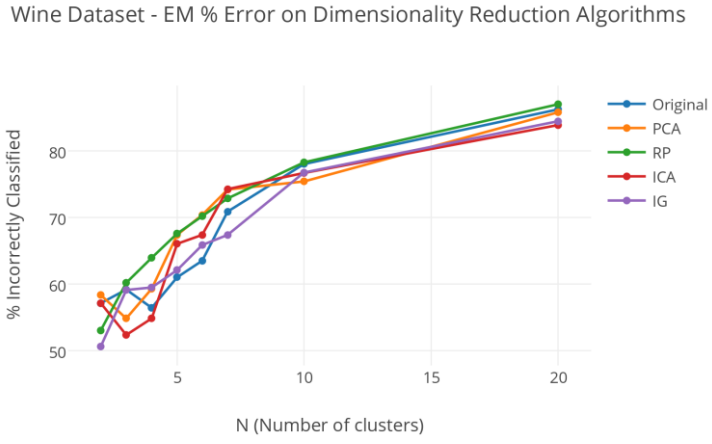
Overall for both datasets, clusters didn't line up well neither naturally nor did they line up well with the class labels. It could be mainly because of the problems I chose because as already know from my previous assignments, Wine dataset is very resistant to showing good results.

Dimensionality Reduction Algorithms

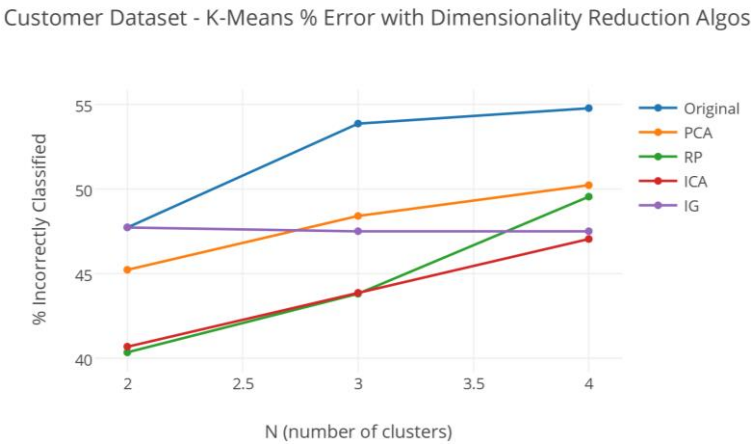
Four dimensionality reduction algorithms were ran on both datasets are:

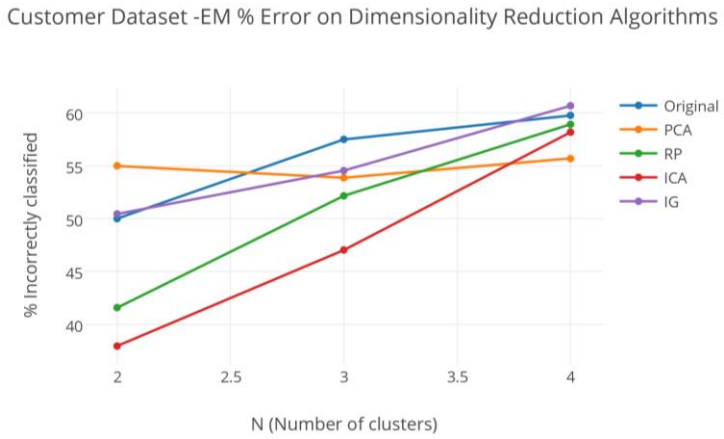
- *Principal Component Analysis (PCA)*
- *Random Projections (RP)*
- *Independent Component Analysis (ICA)*
- *Information Gain (IG).*

The last algorithm I chose was **Information gain**. It evaluates the worth of an attribute by measuring the information gain with respect to the class. Both clustering algorithms were ran after applying all 4 of these techniques on both datasets. The comparison is shown in below graphs along with results from original set of attributes.



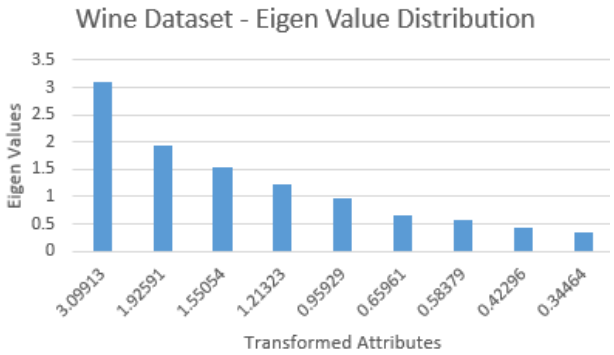
On Wine Dataset, accuracy mostly improved as compared to original dataset though marginally only. Although results varied depending on the number of clusters, but with K-means algorithm, for 2 clusters (which showed best accuracy), ICA and PCA yielded significantly better results as compared to original data. Worst performance was recorded with Random projections on an average. This was expected behavior as all the attributes contribute to wine quality and curse of dimensionality is not really the factor here. Similarly for Expectation Maximization, for two clusters Information Gain did better but for three clusters, again ICA and PCA did better than original data. It can be said that Wine dataset is not a victim of curse of dimensionality and it is just a tough to problem to crack.

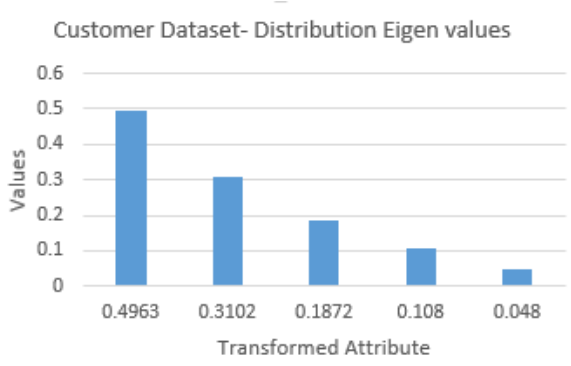




Unlike Wine dataset, Customer Dataset showed promising results and improved the accuracy ranging from 7-13%. ICA and Random Projections gave more accuracy as compared to original data. Both with K-Means clustering and Expectation Maximization.

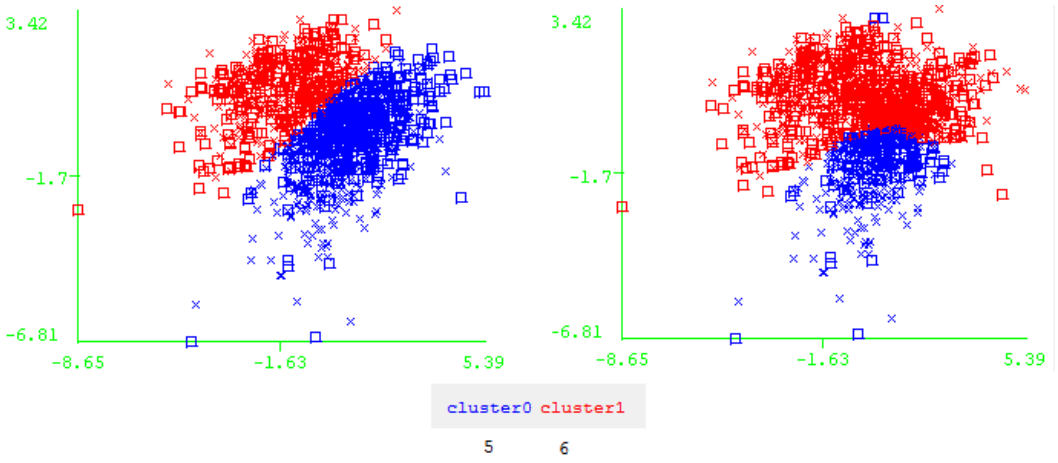
Principal Component Analysis (PCA) : Overall, PCA tells which values are useful as per the Eigen values and its really fast. The Eigen values generated after running PCA on both datasets are shown below



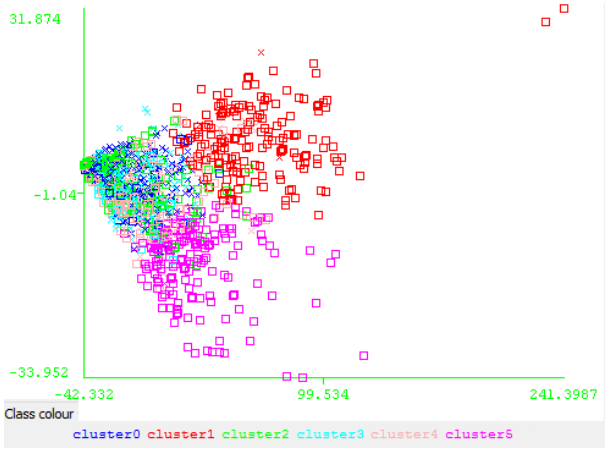


Wine dataset was reduced from 11 to 9 attributes and Customer dataset was reduced from 7 to 5 attributes. I also ran PCA to bring the dimensions down to 2 attributes for both the datasets and saw a difference in accuracy of

about 1-2%, at times it increased and at times it decreased. It must be noted that as expected, the sum of squared error within a cluster was way lower with just 2 attributes. Wine Dataset after clustering into two clusters. It can be seen that close to 50% of attributes in each cluster produced error. This clustering is generated using first two components with maximum Eigen values just to reduce the dimension down to show data visually.



This clustering is done using K-Means in an attempt to see how original labels line up with the newly formed clusters.



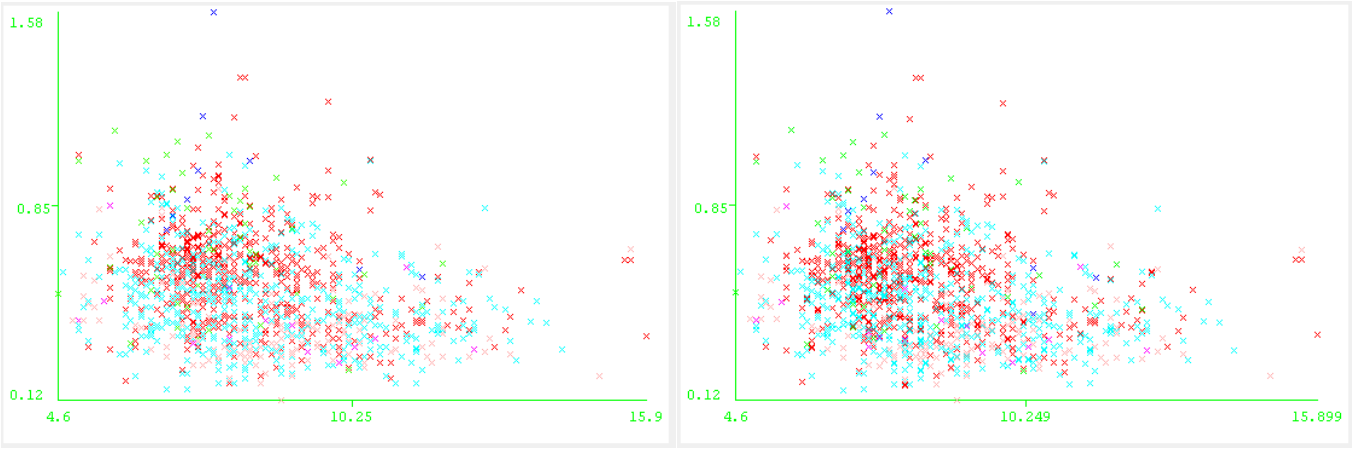
Cluster 0 <-- 5	Cluster 3 <-- 6	Cluster 1 <-- 4
Cluster 4 <-- 7	Cluster 2 <-- 8	Cluster 5 <-- 3

Incorrectly clustered instances: 59.8499 %

Reconstruction using PCA:

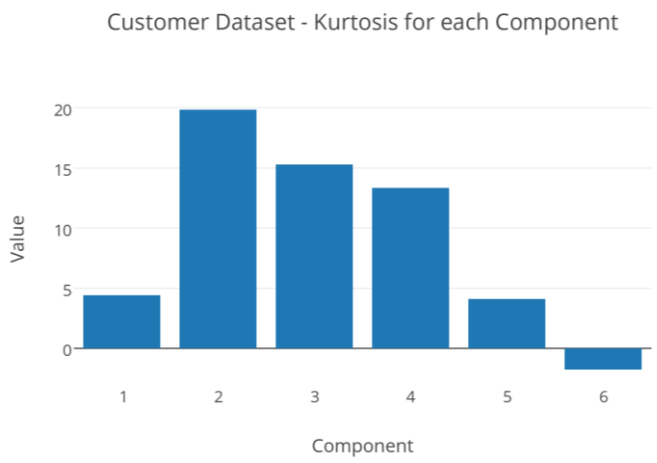
PCA was really good at reconstruction and gave a sum of squared error of:
When reduced and reconstructed from 9 attributes : `4.486210236666676E-4`
When reduced and reconstructed from 7 attributes : `1.1629909691385023`

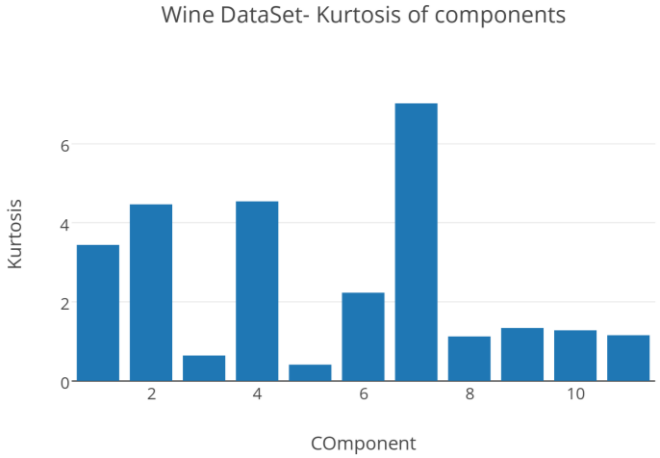
Even visually it can be seen that data was reconstructed very well after PCA as shown below:



Plot of first and second dimension of original dataset and reconstructed dataset after PCA

Independent Component Analysis (ICA): ICA did fairly well on both the datasets as compared to other dimensionality reduction algorithms. As seen below few of the components had high kurtosis values. From Wine dataset, 3rd component, 5th component had close to Gaussian distribution, so I went forward with 9 components for analysis. Wine had some uniformly distributed attributes that generally varied independent of each other and were not very spiky.Customer dataset had spiky peaks in kurtosis for each component as show below and thus was more effective in generating results better than original results and hence better accuracy.

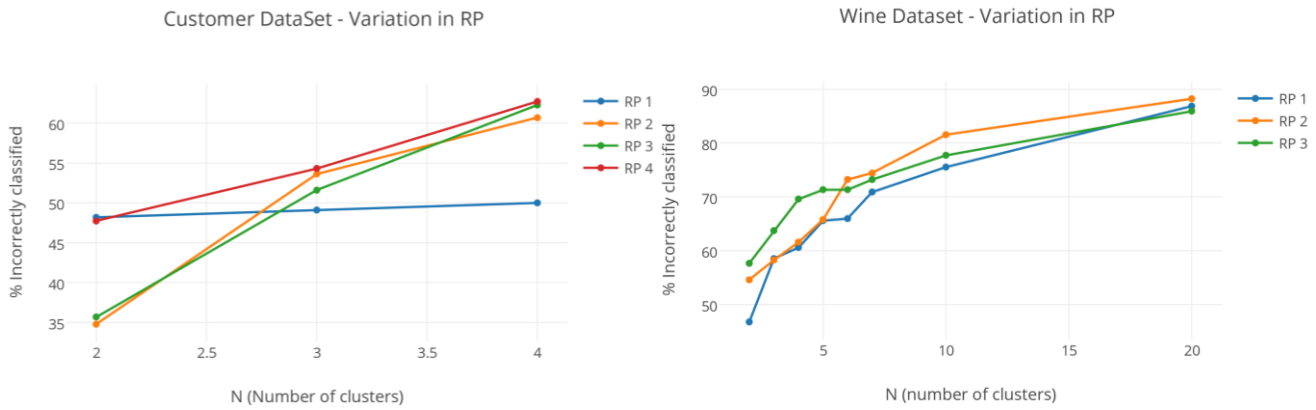




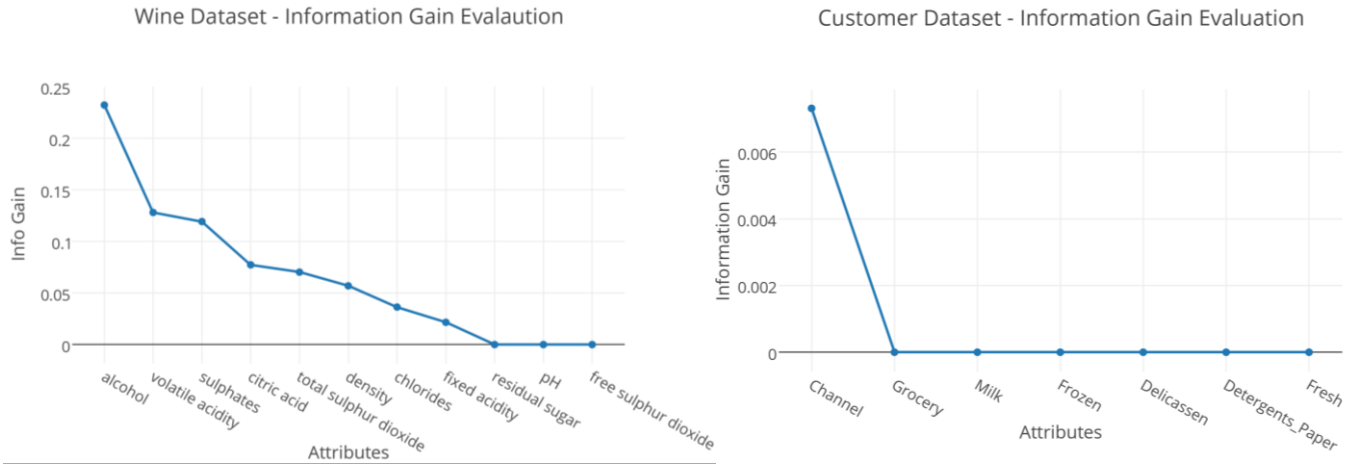
Kurtosis for Wine Dataset was - 2.362076
Kurtosis for Customer Dataset was - 6.209558

Random Projections (RP) : It is really simple and easy method and uses random projection matrices to get lower dimensional spaces. It takes fraction of a second as compared to other algorithms. I ran Random Projection multiple times and each time saw different results. At times, results improved and at times they worsened as expected. When dimensions with high information get picked up at random, accuracy improves. Speed can

further be improved by reducing the number of dimensions to select but at the cost of accuracy. But in one of the experiments I ran, even with two dimensions I got really lucky and got error down to 35% for Customer Dataset and to 45% for Wine Dataset. However taking the average results lowered the accuracy. Reconstruction on RP was fairly poor as compared to PCA and SSE was 302787.9747065846



Information Gain Evaluation (IG) : It shows how much information with respect to class is carried by each attribute. And as seen below For Wine Dataset, only 9 attributes carry information and for Customer Dataset only 1 attribute.

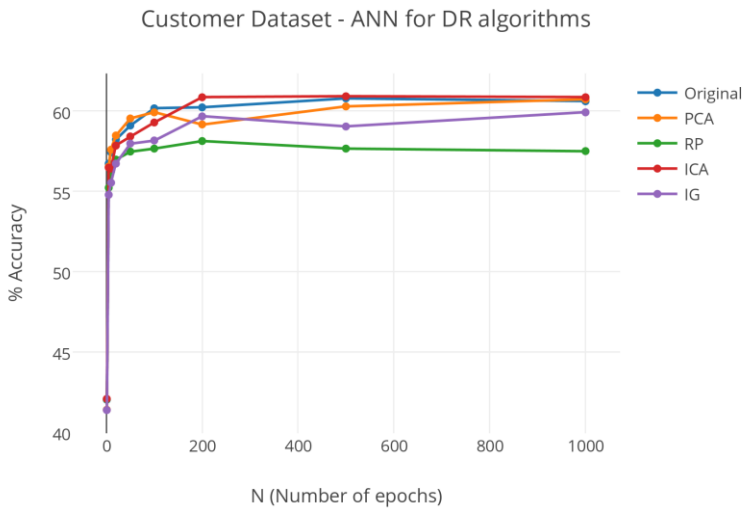


Information gain did best on Wine Dataset when clustered with EM algorithm when I used only top 5 more informative attributes only for two clusters. However as the number of clusters increased, the shortcoming of the algorithm were exposed and information gain was no longer sufficient to reduce dimensions. Reconstruction was not possible with ICA with my currrecnt understanding.

Neural networks:

I used Multilayer Perceptron to implement Neural Networks. As previously found in earlier assignments, Wine dataset was very resistant to neural networks and even with increased number of iterations, changes in momentum or learning rate, it didn't perform well. After applying various dimensionality reduction techniques, I re-ran neural network algorithms and results are shown below. As compared to original data, data reduced using Independent Component analysis performed marginally better with higher iterations and Principal Component

Analysis performed better with fewer iterations. Again, this makes me believe that curse of dimensionality is probably not the reason for poor performance of this dataset.



Clustering algorithms as dimensionality reduction algorithms

This was the most magical experiment because just one attribute, i.e. cluster value captured by K-Means and Expectation Maximization was enough to get same accuracy as was using all the 11 attributes. This shows that clustering algorithms actually captured the information well. Below graph shows how more clusters capture more information and yielded better accuracy. Another interesting thing to notice was that unlike a typical ANN experiment which takes a lot of time and where time taken is directly proportional to number of epochs, information captured by clustering is independent of the number of epochs. Hence we need not run ANN algorithms for more iterations as increasing the number of epochs had no impact on the accuracy. Both 5 iterations and 500 iterations resulted in same accuracy.

Expectation Maximization did slightly better in capturing the information into clusters as compare to K-Means because Expectation Maximization captured the probability of an object belonging to a cluster rather than absolute value.