Assignment 3: CS7641 - Machine Learning

Saad Khan

November 8, 2015

1 Introduction

This assignment covers applications of supervised learning by exploring different clustering algorithms and dimensionality reduction methods. The intent is to compare and analyze these techniques and apply them as pre-processing step to train neural networks. To explore clustering, I used K-Means and expectation maximization (EM) and to highlight dimensionality reduction, I examined principal component analysis (PCA), independent component analysis (ICA), random projections (RP) and Info gain (as feature selection algorithm of choice).

2 Datasets

Both datasets used are chosen from assignment 1 and were taken from the UCI machine learning repository. The Pima Diabetes dataset was used as it is and a subset of the whole wine dataset was used.

2.1 Pima Diabetes Dataset

First dataset contains 768 instances with 8 variables containing information for patients at the Pima Indian community in Arizona. Class labels are associated with the data, of which 268 instances classify as type 2 diabetic (tested positive) and 500 classify as not having diabetes (tested negative). Main constraint during the study was that the patients were all females with at least 21 years of age. This dataset can be very useful for medicinal purposes such as determining medical condition or symptoms for diabetes.

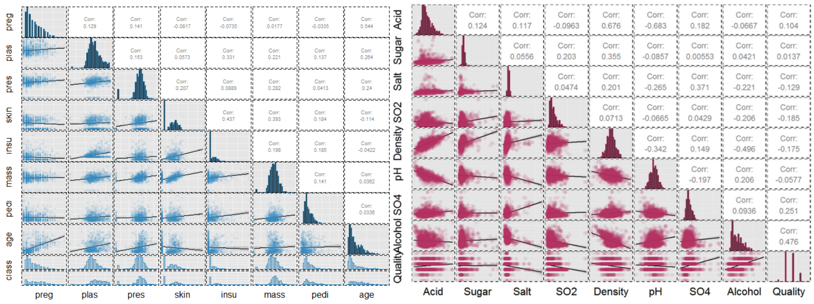

Figure 1: [Correlation Matrix] LEFT : Pima Diabetes, RIGHT : Red Wine

2.2 Wine Quality Dataset

The second dataset is a subset of the whole wine quality dataset used in assignment 1. The dataset originally, has 2 sub-datasets, white wine quality and red wine quality. The datasets have class labels (quality) ranging from 0 - 10 (10 being the best) which I had combined and reduced to 2 for binary classification in assignment 1. Now I have reverted back to the original quality class labels and only selected the red wine quality dataset for this assignment. The reason for selecting this subset was that when I applied clustering algorithms to the whole dataset that I used for assignment 1, clustering was being done based on the color of wine samples rather than quality. This was misleading so I only used the red wine quality dataset. The original class labels associated with this dataset range from 3 - 8 and the dataset has 1599 instances with 11 features. This data can be helpful in the design of intelligent systems that can predict wine quality.

Figure 1 shows the attributes and correlation between them for the 2 datasets. With a few exceptions, most of the features seem to be positively correlated in both datasets which might help in clustering. These datasets have been taken from different areas of study and can be very useful in respective fields of endocrinology and wine making/tasting.

3 Clustering

Clustering is a method of grouping instances in such a way that instances in the same group (cluster) are more similar to each other than to those in other clusters. Here I have explored 2 of these techniques, K-Means clustering and expectation maximization (EM). Weka was used for clustering algorithm implementation along with some help from R. Graphs and plots for the analysis were generated mostly using MS Excel.

3.1 K-Means Clustering

3.1.1 Introduction

K-Means, a centroids based model, clusters attributes into groups by choosing k points at random from the data and setting these to be cluster centers. Then it assigns each point to one of the k clusters based on a specific distance metric such as euclidean, manhattan, etc. It then re-evaluates cluster centers and makes new centroids for the points grouped in this cluster. The algorithm keeps on reassigning cluster centers iteratively to the instances until it converges.

3.1.2 Analysis

First I used SimpleKMeans implementation of K-Means in Weka. For both datasets, I ran the algorithm for increasing values of K using euclidean and manhattan distance metrics. 'Within cluster SS errors' for euclidean and 'sum of within cluster distances' for manhattan were plotted to have general idea about the number of clusters. I tried 2 different random seeds but the outputs were almost similar so I focused on using 1 random seed number.

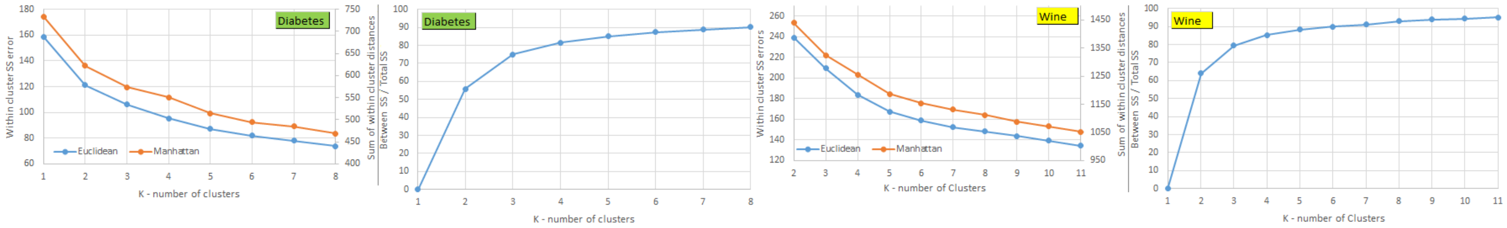

Figure 2: [K-Means] LEFT : Diabetes, RIGHT: Wine

Diabetes ≫ As the diabetes dataset originally had binary class labels, the intuition was to consider k = 2. In order to investigate this further, I compared 2 distance metrics with increasing values of k from 1 to 8. Figure 2 [Outer LEFT] shows the 'with SS error' for the euclidean metric on primary axis and 'sum of within cluster distances' for manhattan metric on the secondary axis. Using the 'elbow methodology' to find most appropriate k, it looks like that k might be either 2 or 3 for both distance metrics.

I also used K-Means in R and plotted ratio of between sum of squares (BSS) to total sum of squares (TSS) for increasing k. BSS/TSS curve, using elbow method, shows k equals 3 in Figure 2 [Inner LEFT]. Comparing the elbows in both these curves for the diabetes dataset, I am certain that k = 3 for both K-Means (Weka and R) implemented for the diabetes dataset.

Wine ≫ For the wine dataset, again I plotted 'with SS error' and 'sum of within cluster distances' on the same graph with increasing number of k from 2 to 11 as shown in Figure 2 [Inner RIGHT]. I tried applying the elbow method to identify k but there did not seem to be a clear cut winner. By looking at the curves, it was hard to pick one value of k but my inclination was towards k = 5.

The BSS/TSS ratio curve for the wine dataset is showing the elbow in a much better way in Figure 2 [Outer RIGHT]. If we compare both curves, the more appropriate choice for number of clusters should be 3. Wine dataset also has class labels associated with it (quality : 3, 4, 5, 6, 7, 8) and with k = 3, clusters do not line up with the labels.

NbClust in R was also used to find the appropriate value for k. NbClust uses 30 indices for determining the no. of clusters and proposes the best clustering scheme from the results (by voting) after trying combinations of no. of clusters, distance metrics and clustering methods. Results using NbClust are shown in Figure 3 [LEFT Section] for K-Means with both euclidean and manhattan distances.

Diabetes ≫ Figure 3 [Outer LEFT] gives most of its votes to k equals 3 which is consistent with what we saw earlier using SimpleKMeans in Weka and KMeans in R.

Wine ≫ Similar results were observed for the wine dataset which can be seen in Figure 3 [Inner LEFT]. Although, k = 2 comes close but most commonly suggested clustering scheme is k = 3.

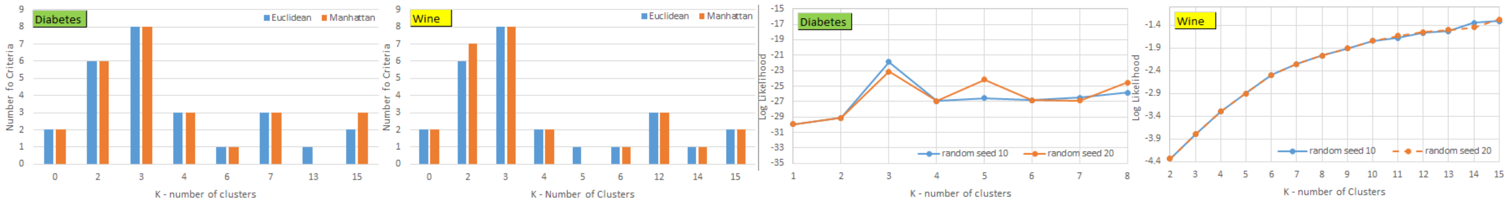

Figure 3: [LEFT] : K-Means using NbClust, [RIGHT] : EM using Log Likelihood

3.2 Expectation Maximization (EM)

3.2.1 Introduction

Expectation maximization (EM) in contrast to k-means, is modeled using probability distributions. It identifies maximum likelihood parameters in cases where equations cannot be solved directly. Initially, EM implementation in Weka was used for the analysis. This implementation has 2 options to find the appropriate number of clusters either it automatically computes k or we choose manually and then follow elbow method to identify k at a good fit point.

3.2.2 Analysis

First I used EM with automatic cluster selection. The number of clusters it gave for diabetes dataset was 3 but for wine dataset it showed a ridiculous number, 17 and 19 with 2 different random seeds that I tried. Then I manually changed the values for k and plotted log likelihood with increasing values of k. Log Likelihood can have +ve or -ve values. Increasing the value of k, normally increases log likelihood. The curve normally flattens out for higher values of k and no. of clusters can be determined by applying the elbow method where it flattens out. The log likelihood plot is shown in Figure 3 [LEFT Section].

Diabetes ≫ Figure 3 [Inner RIGHT] shows few peaks as opposed to continuous increase in log likelihood. After a major peak at k = 3 the curve flattens out so for the 2 random seeds that I tried, I chose k = 3. The automatic cluster selection for EM also gave k = 3, although, it does not comply with the binary classification labels for this dataset but is consistent with the 'k' I got using KMeans earlier.

Wine ≫ In Figure 3 [Outer RIGHT] the log likelihood plot for the wine dataset is much smoother with decreasing gradient for increasing values of k. The curves for 2 different random seeds tried are superimposed on each other and using the elbow method it seems like k = 6. If k is chosen to be 6 then clusters would match the class labels this dataset has.

ClValid package in R was also used to determine values of K using both K-Means and EM. Package uses, 'internal measures' technique to produces plots for connectivity, Silhouette Width and Dunn Index to determine k. Connectivity indicates connectedness of clusters and has a value between 0 and ∞ (should be minimized). Silhouette Width is the average measure of the degree of confidence in a particular clustering assignment and is [-1,1] (1 being well-clustered). Dunn Index is the ratio of the smallest distance between observations not in the same cluster to the largest intra-cluster distance. It has a value between 0 and ∞ (should be maximized). ClValid results are shown in Figure 4 [LEFT Section].

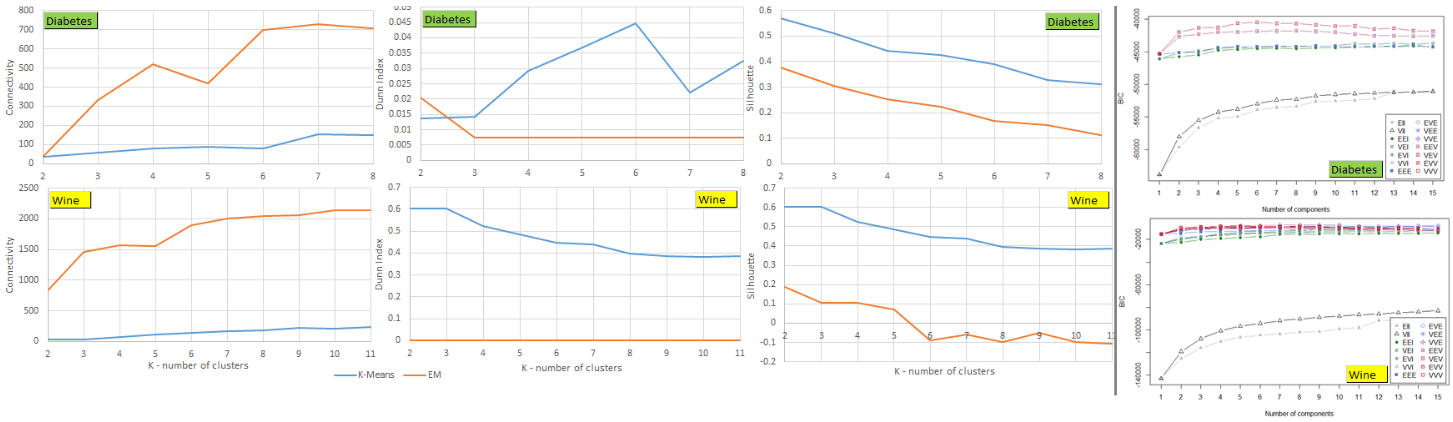

Figure 4: [LEFT Section] : EM and K-Means using ClValid, [RIGHT Section] : EM using Mclust

Diabetes ≫ Using ClValid, for K-Means I observed k = 2 for both connectivity and silhouette & k = 6 for Dunn index. For EM it showed k = 2 for all 3 internal measures.

Wine ≫ In case of wine dataset, for both K-Means and EM, k was equal to 2 for all 3 measuring techniques.

I only used ClValid package in this section and will not be using it from here on as it provided very inconsistent results specially when clustering was applied after dimensionality reduction.

Another package in R, 'Mclust' was also used to determine k using EM based on Bayesian information criterion. Applying it to the datasets, I got k = 6 for the diabetes dataset and 10 for the wine dataset. The MClust plots are in Figure 4 [RIGHT Section].

3.3 Cluster Analysis

3.3.1 Appropriate number of clusters

Combining everything together, I plotted bars for the values of 'k' obtained using different methods in Figure 5.

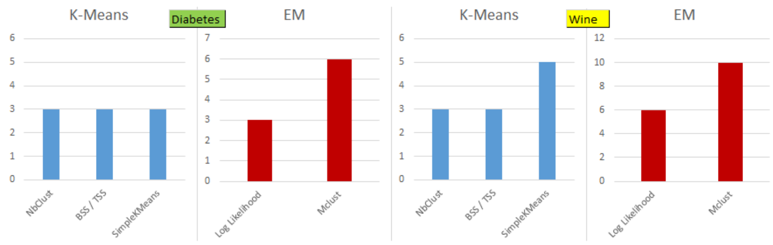

Figure 5: [Clustering Analysis] LEFT : Diabetes, RIGHT : Wine

Diabetes ≫ For the diabetes dataset, all of the methods used suggested k = 3 for K-Means while for EM k = 3 and 6 were suggested. For this dataset I chose the lower value of k = 3 for both clustering algorithms.

Wine ≫ For the wine dataset I also chose the value of k which was lower for both clustering algorithms, i.e. k = 3.

NOTE : I will be using these methods (3 for K-Means and 2 for EM) throughout this report to find the number of clusters when applying clustering after dimensionality reduction.

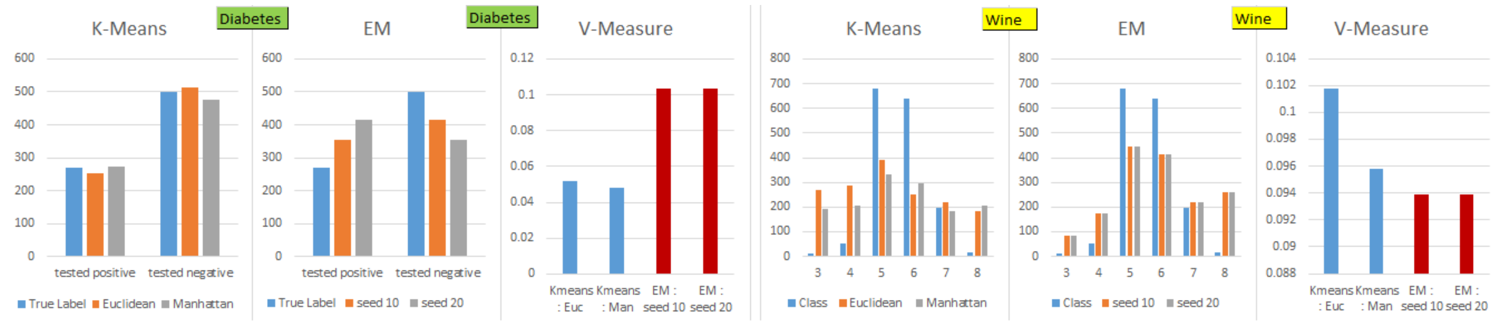

Figure 6: [Cluster Line-up and solution] LEFT : Diabetes, RIGHT : Wine

3.3.2 Cluster Line-up and best clustering solution

In addition to the appropriate number of clusters identification performed above, I also plotted cluster line up in the ideal case if k was equal to the classification labels that are associated with the 2 datasets. Figure 6 shows how would the clusters line up if k was equal to class labels. I tried 2 different distance metrics for K-Means and 2 different random seeds for EM. The only setup where clustering came close to match the class labels was with K-Means for the diabetes dataset.

At first, this seemed to be a better clustering solution but plotting v-measure along with the cluster line up suggested otherwise. It turned out (according to this paper on V-Measure http://www.aclweb.org/anthology/D07-104) the clustering performed by EM, rather than KMeans, is a better solution in case of the diabetes dataset as shown by a higher v-measure score for EM in Figure 6 [LEFT Section]. On the other hand, for wine dataset cluster lineup results after applying EM looked more balanced and comparable to the true labels but by looking at the v-measure it looks like KMeans with euclidean metric provides a better clustering solution.

4 Dimensionality Reduction

Dimensionality reduction algorithms focus on restructuring/pre-processing the input data prior to classifying it with learning algorithms. This section explores four such methods namely, principal component analysis (PCA), independent component analysis (ICA), random projections (RP) and a feature selection algorithm InfoGain (algorithm of choice).

4.1 Principal Component Analysis (PCA)

Principal Component Analysis finds basis vectors that 'best explain' the variance in the data with the first (highest ranked) basis vector (1st PC) best fitting the variance in the data. Following PCs have the same criteria but are orthogonal to each other. Weka's PCA implementation was used for this analysis. The plot below in Figure 7 shows the eigen value distribution (scree plot) for all principal components (PCs) associated with both datasets.

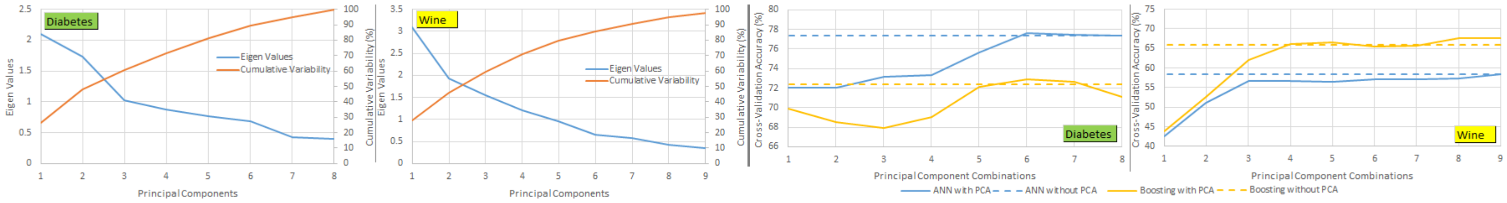

Figure 7: [Scree Plot - Eigen value distribution] LEFT : Diabetes, RIGHT : Wine

4.1.1 Dimensionality Reduction Analysis

Diabetes ≫ Figure 7 [Outer LEFT] shows eigen value distribution and variability for the diabetes dataset. By looking at the curve it looks like first 4 to 5 PCs would fit the most variance in the data as 2 of the highest ranked PCs do not seem to fit the variance in the data that well.

Wine ≫ Eigen value distribution for wine data is shown in the scree plot in Figure 7 [Inner LEFT]. From the look of it first 2 to 3 PCs would be most of the variance in the data.

To highlight this further, I used accuracy curves using 2 classification algorithms (ANN and Boosting with J48 in Weka - this is only for analysis, actual NN training will come later). Each PC was removed from the transformed dataset one by one in ascending order of variance (i.e. last PC then 2nd last PC and so on) and cross-validation accuracy for the learning algorithms was plotted. This trend was compared to the original cross-validation accuracy of the dataset before applying PCA (dotted lines).

Diabetes ≫ For the diabetes dataset, in accordance with what we observed in the scree plot, there is decrease in accuracy for both learning algorithms when we discard the 5th PC as shown in Figure 7 [Inner RIGHT]. By looking at this it seems like 1st 5 PCs would be appropriate for dimensionality reduction.

Wine ≫ For the wine dataset, we do not actually see the decrease in accuracy until we have discarded the 3rd component as shown in Figure 7 [Outer RIGHT], specially in the case of boosting. For this dataset, first 3 PCs would be a better choice to reduce dimensions.

4.1.2 Clustering Analysis

To best capture the variance in the data, I focused on the first 5 PCs for the diabetes dataset and first 3 PCs for the wine dataset and executed the clustering techniques that I ran in the clustering section to determine best value of k. Figure 8 shows the summary for the resulting values of k along with clustering after applying dimensionality reduction (The detailed plots as a result of applying clustering methods after dimensionality reduction are included in the appendix section at the end of this report).

Diabetes ≫ In Figure 8 [LEFT], all of the clustering methods used, suggested k = 3 for K-Means. Bar chart shows values of k generated using K-Means for the diabetes dataset in Figure 8 [LEFT]. Although, this value of k does not line up with the binary class labels for this dataset but cluster diagram using K-Means and 1st/2nd PCs clearly show the distribution of the clusters. Same plot also shows the results of methods used to find k using EM, suggesting k to be 3 and 8. I tried both of these one by one and the most clean and naturally divided clusters were seen when using k = 3 as shown in the lower cluster diagram for 1st/2nd PCs in Figure 8 [LEFT].

Wine ≫ Similar methodology was applied to the wine dataset as well, using K-Means and EM and the results are shown Figure 8 [RIGHT]. All of the methods suggested k = 5 for K-Means. I applied this clustering to the transformed wine dataset using K-Means and 5 clusters can be seen in Figure 8 [RIGHT - upper cluster diagram]. For EM, the value of k = 6 gave distinct clusters which can be seen in the cluster diagram using 1st/2nd PCs in Figure 8 [RIGHT - lower cluster]. This is consistent with the 6 class labels for this dataset.

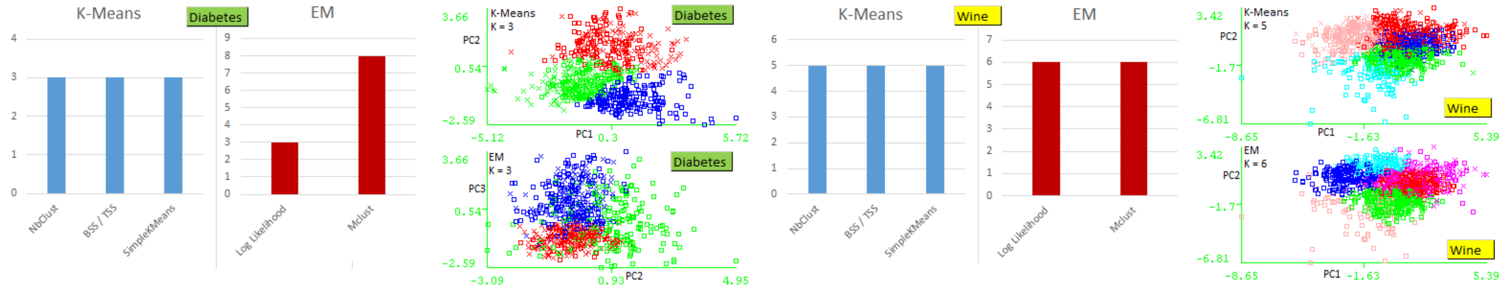

Figure 8: [PCA - Number of Clusters] LEFT : Diabetes, RIGHT : Wine

4.2 Independent Component Analysis (ICA)

Independent Component Analysis focuses on restructuring the input data by increasing the separation between each of the components from one another by finding basis vectors that are independent components of the original data.

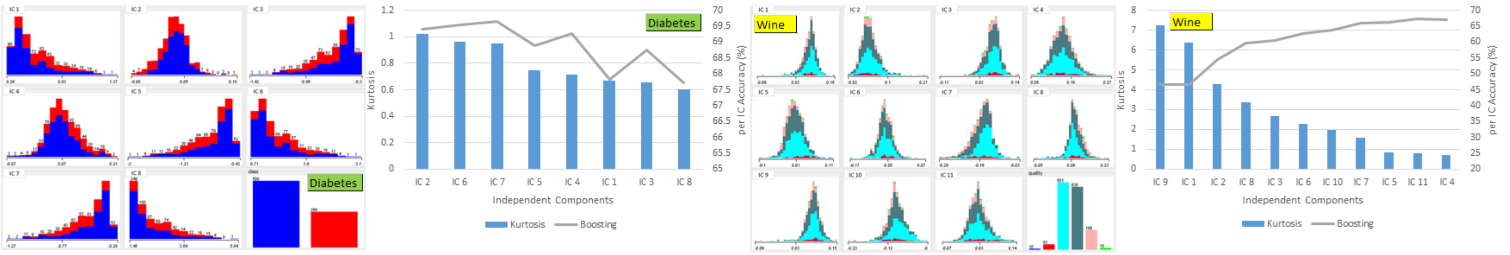

Figure 9: [ICA - Distribution and Kurtosis] : LEFT : Diabetes dataset, RIGHT : Wine dataset

ICA implementation used in this report was 'FastICA' module in Weka. Histogram distribution of the components along with respective kurtosis distribution show extent of non-Gaussianity which is shown in Figure 9. Components are sorted from highest to the lowest kurtosis values (L to R). Kurtosis values were calculated using Excel which generates excess kurtosis by default i.e. kurtosis is already adjusted in a way that for a Gaussian distribution it is zero. The kurtosis plot also includes the cumulative cross-validation accuracy results of Boosting (only for analysis purposes, eg: The accuracy associated with IC 5 for diabetes dataset is actually the accuracy for IC 2, IC 6, IC 7 and IC 5 combined). Using this way of plotting cumulative accuracy on top of kurtosis, I tried to reduce dimensions by identifying and eliminating Independent components (ICs) that had insignificant effect and could be considered as noise.

4.2.1 Dimensionality Reduction Analysis

Diabetes ≫ For this dataset, after applying ICA some of the resulting features exhibited non-Gaussian nature to a certain extent which is depicted by the kurtosis distributions in Figure 9 [LEFT]. Kurtosis values are sorted from left to right and only 3 components have values above 0.8. Along with that the cross-validation accuracy curve for boosting with J48 on the secondary axis shows that components with kurtosis less than 0.8 do not really contribute and in fact accuracy starts to drop when moving from left to right as more components with low kurtosis are added to the dimensionally reduced dataset. Although, none of the components have high kurtosis to suggest high degree of non-Gaussianity but in this case the only meaningful components seem to be IC 2, IC 6 and IC 7 while rest of them can be considered as noise and discarded.

Wine ≫ Wine dataset had more features compared to the diabetes dataset, hence more ICs. By looking at the histogram distributions (Figure 9 [RIGHT]), most of the components seem to have highly leptokurtic distributions, which is also confirmed by the kurtosis distribution, with 3 of highest kurtosis components having kurtosis above 4. Here the accuracy curve for boosting keeps on showing improvement with additional components being added till we see no further improvement when components being added to the dataset have kurtosis less than 1. For this dataset, it seems that all components are meaningful except IC 5, IC 11 and IC 4 which can be discarded as they have kurtosis close to zero.

4.2.2 Clustering Analysis

Diabetes $\gg$ After selecting the components that I wanted to keep for my analysis (ones with kurtosis higher than 0.8) I ran few of the clustering methods to determine the best value for k. The results of these methods is captured by the bar plot in Figure 10 [LEFT] and detailed plots are in the appendix section. No. of clusters suggested by K-Means was 3 for all the methods tested. Cluster distribution for $k = 3$ is shown in upper cluster diagram in Figure 10 [LEFT] using 1st and the 5th IC. Methods used to evaluate k using EM suggested $k = 4$, the cluster diagram for which is also shown in Figure 10 [LEFT - lower cluster].

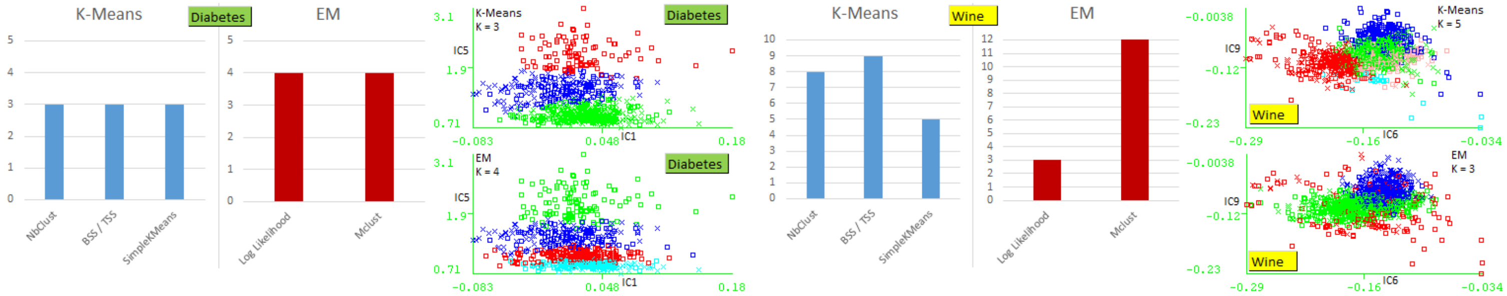

Figure 10: [ICA - Number of Clusters] : LEFT : Diabetes dataset, RIGHT : Wine dataset

Wine $\gg$ Cluster suggestions for the wine dataset were not as straight forward. For both K-Means and EM, I tried the different suggested values of k one by one and ultimately decided to use the lowest values of k. Using K-Means, I chose $k = 5$ and with EM I chose $k = 3$ from the multiple outputs given by the different methods used. Cluster diagrams in Figure 10 [RIGHT] show the results in 2D for 2 of the components.

4.3 Random Projection (RP)

Random projection (RP) is another dimensionality reduction method used to project 'n' total attributes in to k-dimensional space where $k \ll n$. Main benefit of projecting the data this way is to lower dimensions and help save computation cost. Depending on the random projection, the accuracy of the data may or may not be affected that much. The model used for random projection in this assignment is the one implemented in Weka. As the name suggests, I used 5 different random seeds (10, 20 , 30 , 40 and 50) to generate different sets of random components. In order to satisfy the $k \ll n$ condition, initially the no. of components I selected were 1 less than the total features for that dataset. Analysis regarding the best performing component combination is covered in the section below. Again, I used ANN and Boosting with J48 to highlight the best performing component combination out of the 5 random projection subsets. One by one I removed the components in sequential ascending order (eg: for components K1 to K10, removing K10 then K9 an so on) I plotted accuracy to identify the least amount of components required to best reconstruct the data.

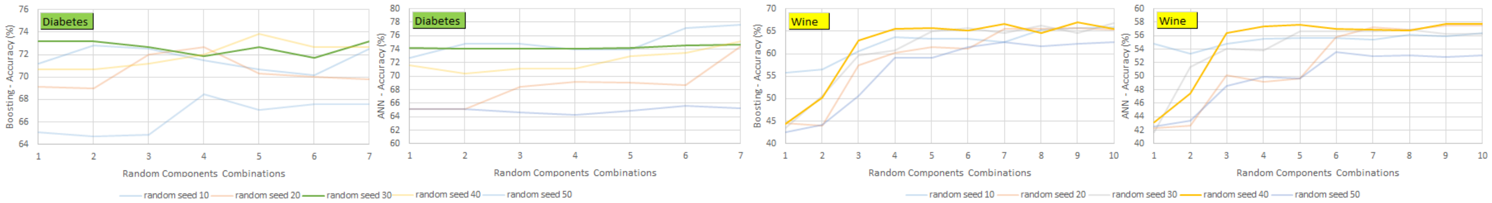

Figure 11: [RP - Best randomly projected subset] : LEFT : Diabetes dataset, RIGHT : Wine dataset

4.3.1 Dimensionality Reduction Analysis

Diabetes $\gg$ For this dataset, the most consistent performing RP component set was the one with random seed 30 (highlighted in green). As it can be seen in Figure 11 [LEFT Section], for both learning algorithms, this set performs quite well. I chose first 5 random projected components out of the total 7, because considering 6th and the 7th component triggers over-fitting.

Wine $\gg$ For the wine dataset, as highlighted with yellow curves in Figure 11 [RIGHT Section], RP components for random seed 40 gave the best performance. Performance starts to drop after the exclusion of the 3rd random component. To maintain considerable level of performance, I chose 1st 5 components to represent the dimensionality reduced set.

4.3.2 Clustering Analysis

Following the selection of the best possible random components, I used some of the clustering schemes to find the best value of k to represent the dimensionally reduced data.

Diabetes $\gg$ The 2 clustering schemes suggested by the K-Means clustering methods were $k = 3$ and 4. I chose the smaller of the 2. Although, not that clean , but one of the possible 2D representations is shown in the cluster diagram Figure 12 [LEFT - upper

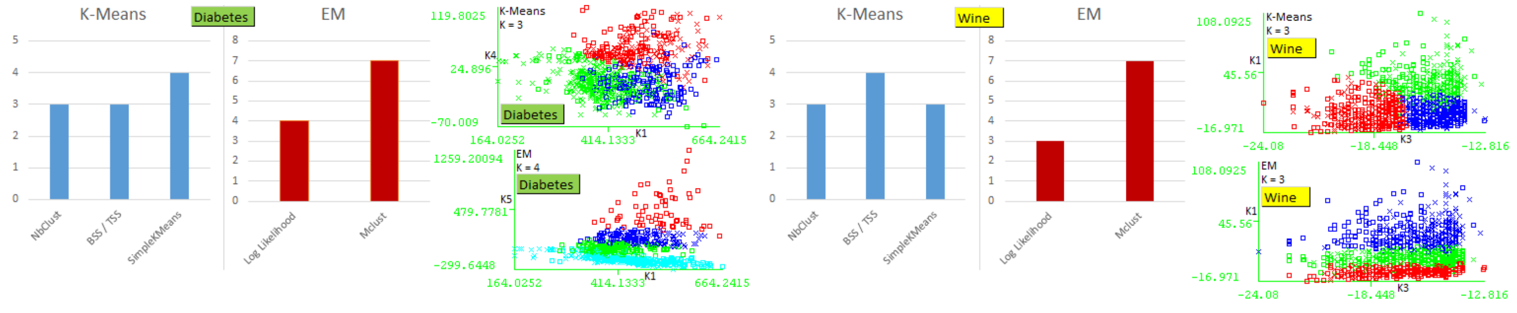

Figure 12: [RP - Number of Clusters] : LEFT : Diabetes dataset, RIGHT : Wine dataset

cluster] using 2 of the random components. Applying EM to the dataset, again I chose smaller of the k values i.e. k = 4. The lower cluster diagram in Figure 12 [LEFT] shows a much cleaner 2D representation using 2 random components.

Wine ≫ Clustering results after applying K-Means and EM to the dimensionally reduced wine dataset are shown in Figure 12 [RIGHT Section]. For both clustering algorithms the smallest value of k was used, i.e k = 3. One of the many cluster 2D representations for these are also shown here.

4.4 Info Gain Attribute Evaluation (IG)

InfoGain attribute evaluator is an internal Weka attribute selection scheme which evaluates the worthiness of an attribute by measuring the information gain with respect to the class.

$$\textbf{InfoGain(Class,Attribute)} = \textbf{H(Class) - H(Class — Attribute)} \tag{1}$$

InfoGain works on the same principal using which decision trees evaluate information gain to determine which attribute goes on top node for best splitting the tree and so on. For both datasets this feature selection algorithm has ranked attributes based on their information gain as shown in Figure 13. Based on the ranking, I sequentially dropped the attributes one by one in ascending order from (eg: dropping 'pres' then 'pedi' from the diabetes set and so on) the dataset running NN and Boosting to note down where the performance deteriorates. Solid line in the curve indicates the cross-validation accuracy when a ranked attribute is dropped from the dataset and the dotted line indicates the cross-validation accuracy for the original dataset.

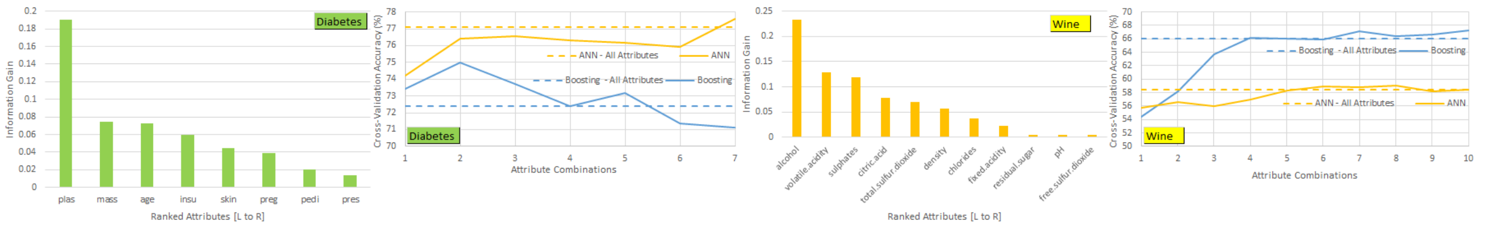

Figure 13: [IG - Ranked Attributes] : LEFT : Diabetes dataset, RIGHT : Wine dataset

4.4.1 Dimensionality Reduction Analysis

Diabetes ≫ For this dataset, it is observed that running boosting actually helped to improve accuracy when I discard lowered ranked attributes while for ANN it almost never changes till the 2nd highest ranked attribute is removed. By looking at the accuracy curves in Figure 13 [LEFT Section], only 2 attributes would suffice for this dataset, i.e. 'plas' and 'mass'.

Wine ≫ Similar setup was run for the wine dataset as well and ranked attributes can be seen in Figure 13 [RIGHT Section]. The accuracy plots for ANN and Boosting show that both algorithms performed well up till the point where 5th highest ranked attribute was excluded form the dataset. Based on that I kept top 5 ranked attributes in the feature reduced dataset.

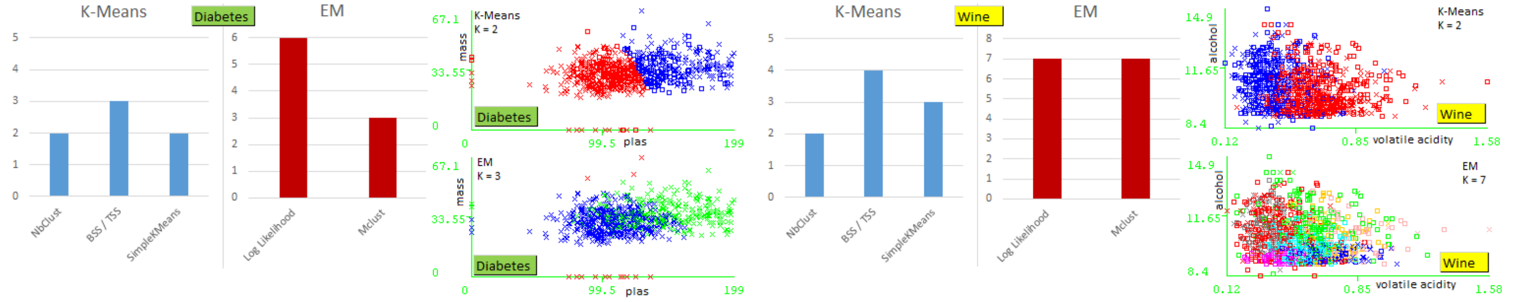

Figure 14: [IG - Number of Clusters] : LEFT : Diabetes dataset, RIGHT : Wine dataset

4.4.2 Clustering Analysis

After I had determined the reduced subset of features, I then applied the clustering techniques to determine best value of k.

Diabetes ≫ For the diabetes dataset, 2 out the 3 methods used to determine k using K-Means suggested k = 2. On the other hand, 2 methods that were used to determine k using EM suggested either k = 3 or k = 6. I chose the lowest of the 2. Cluster diagrams in Figure 14 [LEFT] show the diagrammatic representation.

Wine ≫ Again for this dataset, I chose the lowest value for k using K-Means, i.e. k = 2. On the contrary, both clustering methods using EM suggested k = 7. Cluster representation is shown in Figure 14 [RIGHT].

5 Neural Network Performance

5.1 Dimensionality Reduction and Neural Network

5.1.1 Introduction

After applying clustering and dimensionality reduction to the 2 datasets, I chose the wine dataset to further train a neural network and analyze its performance with and without dimensionality reduction.

5.1.2 Implementation

Neural network used for analysis was multi-layer perceptron implementation in Weka. For the purpose of comparison, I trained the neural network using the original dataset and used it as reference. The hidden nodes (single layer - 8 nodes) for the neural network and various tuning parameters such as learning rate (0,3), momentum (0.2) and epochs (500) were kept constant in order to fully analyze if dimensionality reduction was beneficial or not and if beneficial then to what extent was it beneficial.

5.1.3 Procedure

I started of by passing the dimensionally reduced datasets (in this case 4) with all components to the neural network and then kept on passing further reduced dataset by eliminating lower ranked components one by one. This is the same exercise I did in order to determine the least number of components in the dimensionality reduction section above.

5.1.4 Analysis

Plots in Figure 15 show the original dataset accuracy as a result of classification performed using neural network (red dotted line for reference). The blue curve in each plot shows the accuracy when different combinations of the components were passed to the same neural network. Yellow markers on blue curves indicate the number of components that were used during dimensionality reduction phase to create clusters for this dataset.

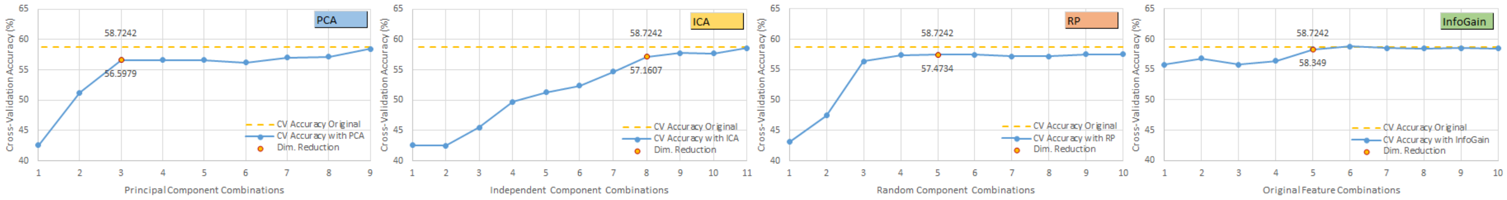

Figure 15: [Neural Network Accuracy for Wine Dataset] L to R : PCA, ICA, RP and InfoGain

Components required for the accuracy to remain almost at par with the original dataset accuracy for each of these algorithm was different. It can clearly be seen that PCA required least number of components, (i.e only 1st 3 PCs) in order to maintain good accuracy while for other 3 algorithms, more components were required to maintain adequate accuracy. The original dataset obviously performs better, but its important to highlight that PCA runs at a significantly faster rate for a small trade-off in accuracy as in this case for less than 2% decrease in accuracy, computation cost came down by almost 30%.

Figure 16 compares the performance of all 4 reduced datasets that were used in the previous section for clustering after applying dimensionality reduction (indicated by the yellow markers in Figure 15). By looking at the plot, it was observed that, although, applying PCA reduced the size of the dataset but in terms of computation time datasets reduced after random projections and InfoGain analysis were not far behind with difference less than 0.2 seconds. In fact after applying InfoGain, performance is better than other algorithms for all performance metrics including precision, recall and f-measure as seen in Figure 16 [RIGHT].

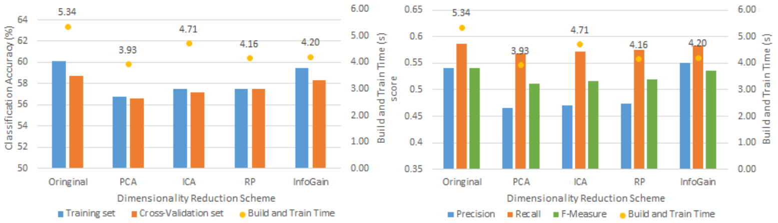

Figure 16: [Performance Comparison after dimensionality reduction and Neural Network training]

5.2 Clustering and Neural Network

5.2.1 Introduction

This section highlights, how performance of the dataset varies if clusters are introduced as attributes.

5.2.2 Implementation

I covered 2 scenarios in this analysis, 1. a subset consisting of original attributes plus clusters as an additional attribute was passed to the neural network and performance was noted and 2. only clusters were used as inputs to the neural network and results were analyzed. I used SimpleKMeans and EM implementations in Weka for this purpose.

5.2.3 Procedure

For increasing numbers of clusters 'k' from 2 - 10 I ran clustering on the wine dataset. Using SimpleKMeans with euclidean and manhattan distance metric I created 4 new scenarios, one with original attributes + clusters [total attribute columns = 11 + 1] for euclidean metric and then another clusters only [total attribute columns = 1] for euclidean metric. I repeated the same with manhattan metric. Then using EM I created 4 similar scenarios with 2 different random seeds.

Figure 17 [LEFT] shows how, for changing values of k, the accuracy varies for the datasets which include both original attributes and clusters. It can clearly be seen that K-Means with manhattan metric for k = 9 performs the best, far better than the original dataset and also takes 0.4s less time to compute.

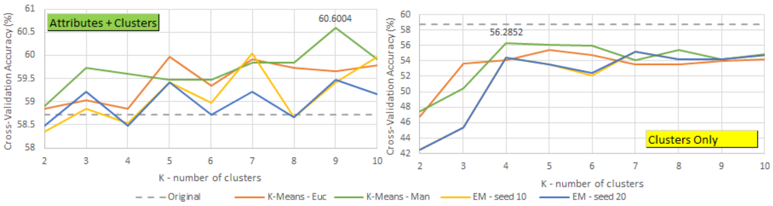

Figure 17: [Neural Network Accuracy when using clusters as attributes]

On the other hand if we look at Figure 17 [RIGHT], none of the scenarios perform better than the original dataset, however, again using manhattan metric, accuracy with k = 4 comes really close within 2.5% of the original dataset accuracy. With no original attributes and only clusters, this dimensionally reduced dataset takes almost half as much time to compute compared to the original dataset. Although, the overall performance of the dataset is not good as it is only classifying 58% of the data but using dimensionality reduction with clusters can save a lot of time with only a minor dip in accuracy. This technique could be very useful for datasets with instances up to 10k and above as it would take orders of magnitude less time to compute at a reasonable level of accuracy. Other performance metric comparison is shown in Figure 18 with accuracy comparison on LEFT and precision, recall, F-measure comparison on RIGHT.

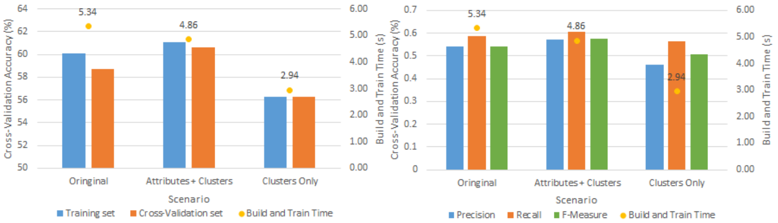

Figure 18: [Performance Comparison after clustering and Neural Network training]

6 Conclusion

In conclusion, dimensionality reduction and clustering algorithm serve the main purpose of decreasing the processing time of algorithms. In the specific case of our datasets, it does not seem to be the case that accuracy was improved, but there were cases in which performance remained virtually the same, whilst running significantly faster.

7 Appendix

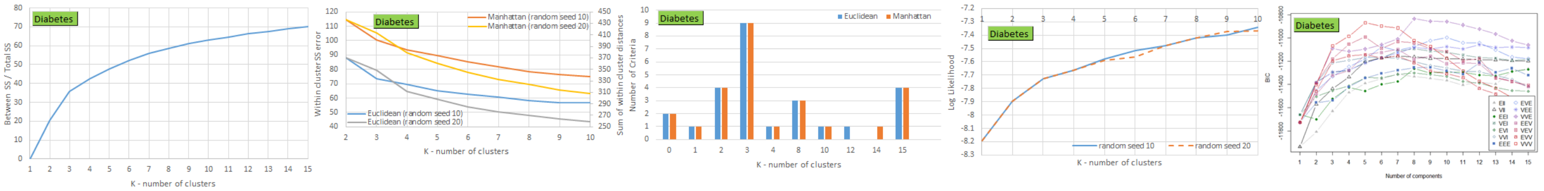

Figure 19: [PCA - Diabetes]

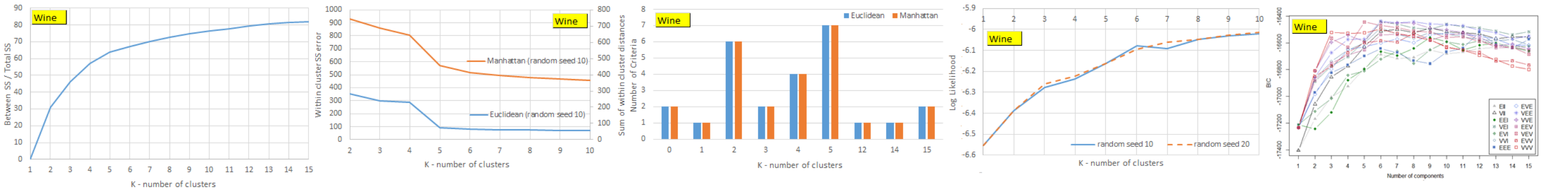

Figure 20: [PCA - Wine]

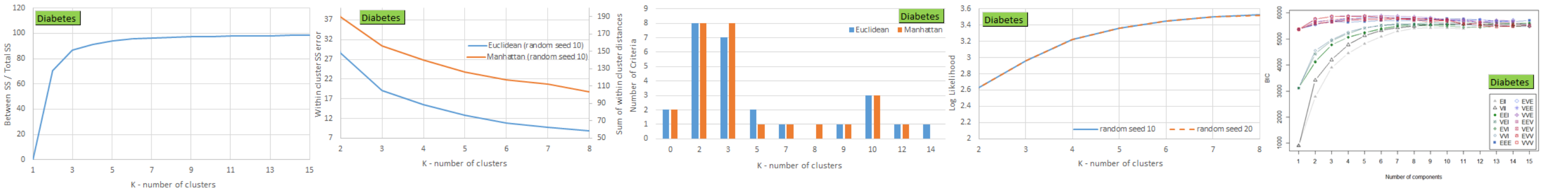

Figure 21: [ICA - Diabetes]

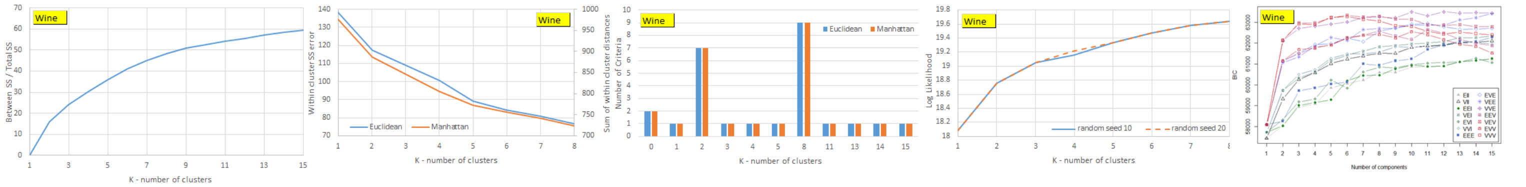

Figure 22: [ICA - Wine]

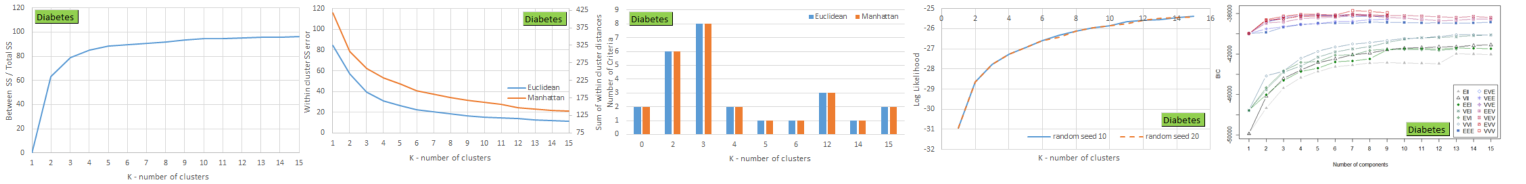

Figure 23: [RP - Diabetes]

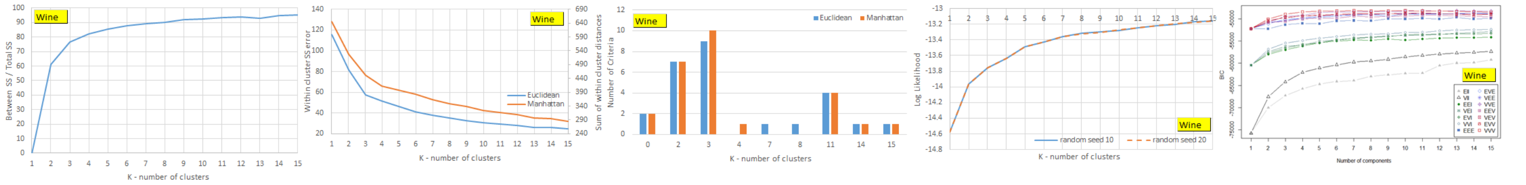

Figure 24: [RP - Wine]

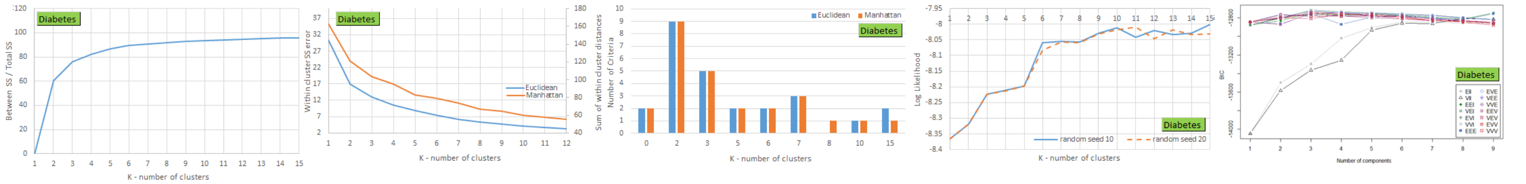

Figure 25: [RP - Diabetes]

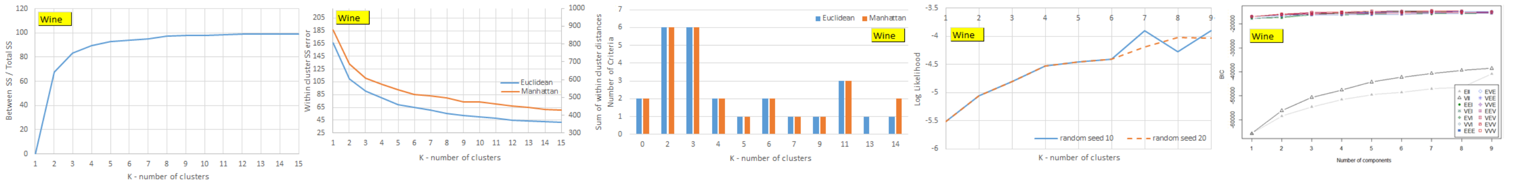

Figure 26: [RP - Wine]