Name: Heng Chang Rong Kelvin
Course: CS7641
Assignment: 3

Unsupervised Learning & Dimensionality Reduction Analysis

Introduction

The goal of this assignment is to analyse the result of various clustering and dimensionality reduction algorithms on two datasets. In this project, I used the two datasets mentioned in assignment one.

Dataset

Steel dataset

The dataset comprises various quantitative metrics about steel plates. It is a binary classification problem to predict the dirtiness level of steep plates, allowing a wide use of applications in many industries.

Contraceptives dataset

The contraceptives dataset helps to determine whether contraceptives are being used based on many attributes of the husband and wife. Modelling birth rates, sales of contraceptives and healthcare related applications are some of the interesting use cases from solving this dataset.

Preprocessing

The preprocessing method for this assignment involves methods such as:

1) Using sklearn MinMax scaler to scale data between 0 and 1
2) Use one hot encoding to encode categorical variables (if any)

Methodology and metric

We use the sklearn library and the following metrics for analysis in this assignment.

Algorithm	**Metric**
K-Means	Silhouette score V measure Sum of squared distances
Expectation Maximization	AIC BIC
PCA	Explained variance Minimum singular values

ICA	Reconstruction error Average kurtosis of feature distributions
Random Projections	Reconstruction error Average kurtosis of feature distributions
Isomap	Reconstruction error
Neural Network	AUC

Part 1 - Clustering

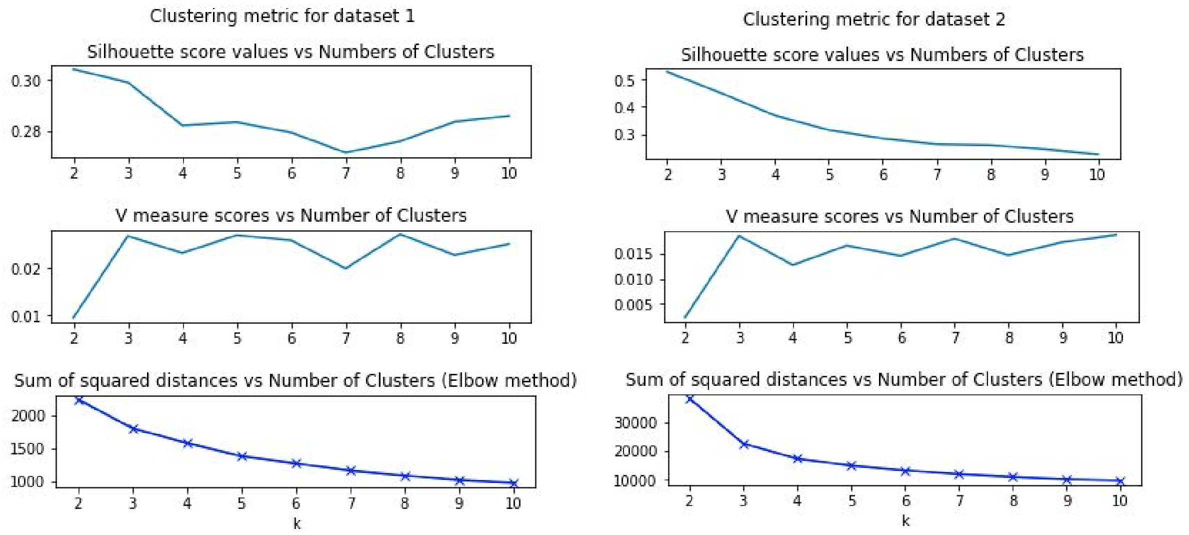
We run two clustering algorithms, K-Means and Expectation Maximization on the two datasets.

K-Means

The K-Means algorithm attempts to segregate the data into K clusters via minimization of sum of squared distances within a cluster. The algorithm can be summarized into the following:

1) Initialize the cluster centroids.
2) Assign every sample to their nearest cluster centroid.
3) Calculate the cluster centroids based on the samples assigned to a cluster.
4) If not converged, go back to step 2.

We use the Silhouette score to measure how similar an object is to its own cluster as compared to other clusters. The V-measure, which is the harmonic between homogeneity and completeness is used to ensure that the cluster labels do not contradict the labels. Lastly, we also show the optimization metric for the algorithm, the sum of squared distances (SSD).



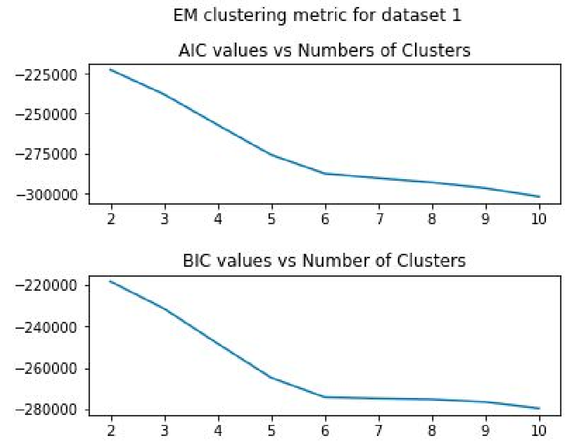
The above are the clustering metric plots generated using K-Means for the steel plates and contraceptives dataset. Both of the datasets have only two different ground truth labels. However, if we try to take the best tradeoff between Silhouette score, V-measure score and

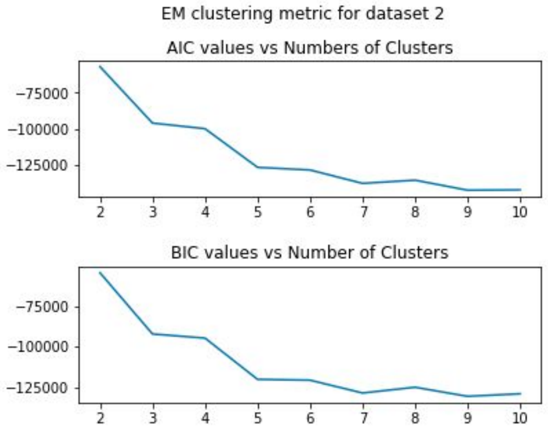
SSD using elbow method, K-Means should assign 3 clusters to both datasets. Perhaps the features in both datasets are not immediately separable by just their features' euclidean distances alone.

Expectation Maximization

K-Means is a special case of Expectation Maximization (EM) when there are hard assignments of cluster labels. EM tries to fit a mixture of gaussian distribution by assigning every sample probabilities of belonging to every cluster. The parameters are then generated to maximize the likelihood of the data given those samples and clusters.

We use the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) to determine the number of optimal clusters. AIC penalizes the larger number of parameters in the model and lower log-likelihood of the model. BIC is very similar to AIC, except that it penalizes the model complexity more.





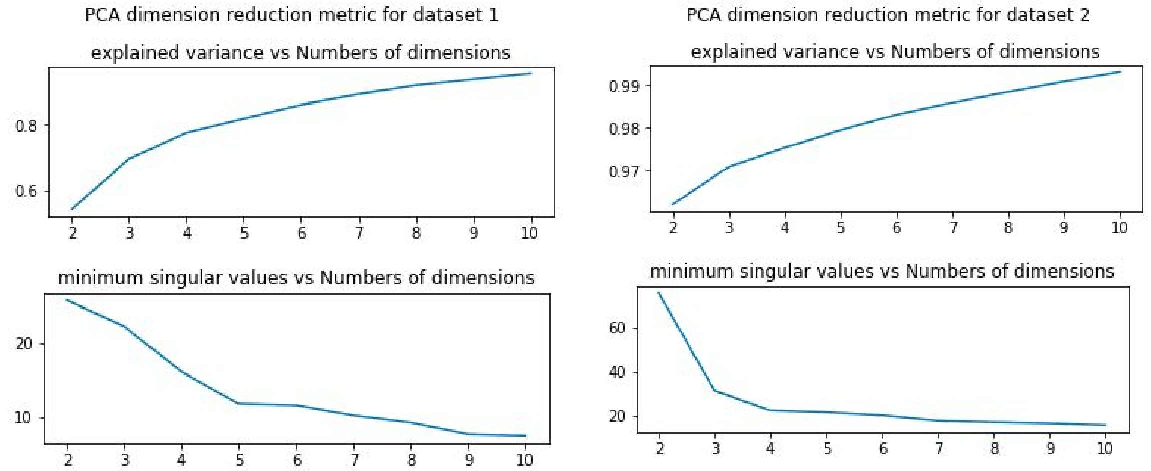
The above are the clustering metric plots generated using EM for the steel plates and contraceptives dataset. Since we are aiming for a lower AIC and BIC value, it is clear that we can choose 6 and 5 clusters for steel plate and contraceptives dataset respectively. Similar to the case of K-Means, the number of clusters does not reflect the number of classification classes in both datasets. Hence, we can also conclude that the features are not really well represented by the gaussian distributions.

Part 2 - Dimensionality Reduction

In this section, we run 4 dimensionality reduction algorithms (PCA, ICA, Random Projections, Isomap) on both datasets.

Principal Component Analysis (PCA)

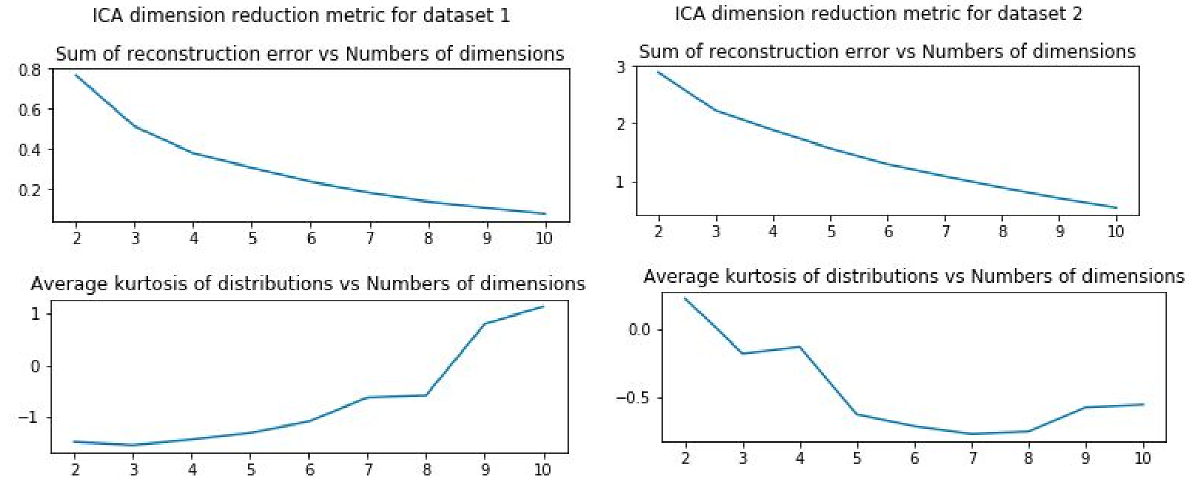
PCA uses an orthogonal transformation to convert multidimensional features into a set of lower dimensional features with linearly uncorrelated variables, which are called principal components. The principal components are ranked from highest to lowest in terms of the power to reconstruct back the original dataset. This method is extremely sensitive to the scale the features, hence we did minmax scaling before utilizing PCA.



The above charts are the results from applying PCA on both datasets. The explained variance is the total amount of variance of the original dataset explained by the chosen principal components after applying PCA. In addition, we also included the minimum of the PCA's singular value to indicate how much marginal impact the last principal component has. We can clearly see that the explained variance chart for the steel plates dataset is increasing well, hence 10 dimensions was chosen. However, we can see that the explained variance for the contraceptive dataset already reached 97% with 3 principal components. It is also further supported by the elbow chart of the singular values. The reason that 3 principal components can easily represent the original features of the contraceptive dataset is because PCA simply tries to recombine the one hot encoded features of the dataset into 3 powerful principal components.

Independent Component Analysis (ICA)

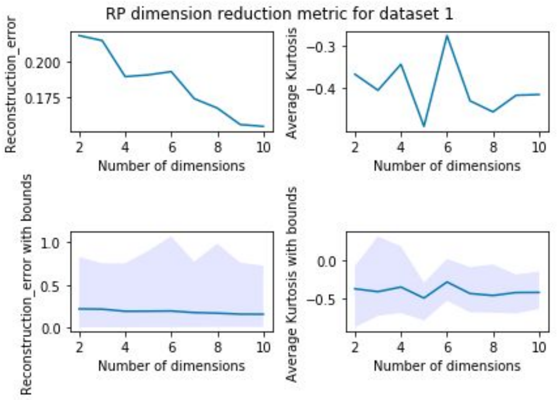
ICA is typically used in signal processing to separate a multivariate signal into additive subcomponents. The underlying assumption is that these subcomponents are non-Gaussian distributed and are independent from one another.

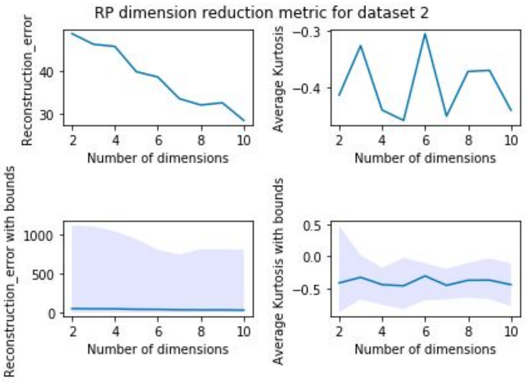


The above charts are the results from applying ICA on both datasets. The reconstruction error and average kurtosis of the distributions of new features were plotted against the number of dimensions. Since the dimensions of our datasets are not high, we stopped the charts at 10 dimensions. As we can observe from the charts, there is no clear elbow line to choose the optimal dimensions. Hence, we chose 10 dimensions for both datasets. The reason may be due to the fact that our contraceptives dataset has variables that are non-independent. Hence, PCA is able to generate 3 powerful principal components to represent the original features, whereas ICA is not able to do so.

Random Projections (RP)

RP reduces the dimensionality by projecting the features on a randomly generated matrix. This matrix is sampled from a gaussian distribution and the method preserves the distances between any two data points.

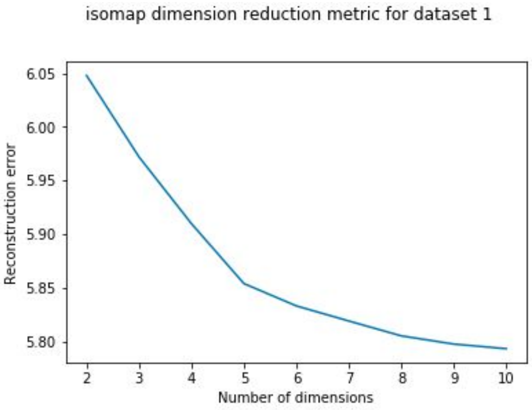


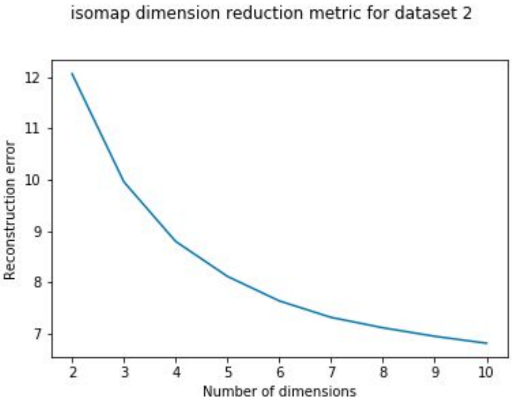


The charts above show the results from applying RP to both datasets. The only difference between charts on the first and second row is to plot out the variation in reconstruction error and kurtosis for 10 trials for every dimension analysed. It doesn’t seem that RP is able to find an optimal number of dimensions for reduction with good kurtosis and reconstruction error metric. However, i will choose to reduce to 10 dimensions for both datasets.

Isomap

Isomap is a nonlinear dimensionality reduction algorithm. It computes the geodesic distance between a point and its neighbours and results in a low-dimensional embedding of a high-dimensional dataset. The intuition is that each neighbour should be closely related in a high dimensional space. Using geodesic distance instead of euclidean distance will preserve this information which is extremely valuable for non-linear data.



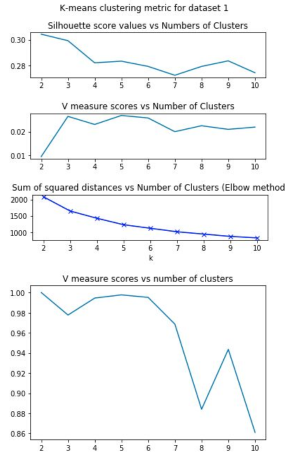


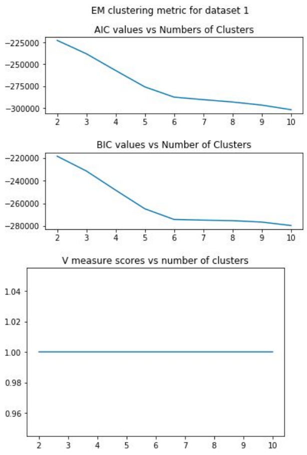
The above charts show the results from applying Isomap to both datasets. The only metric chosen to evaluate Isomap is reconstruction error. It is clear that both charts show a distinct elbow shape. I will choose to reduce to 5 and 6 dimensions for steel plates and contraceptives dataset respectively.

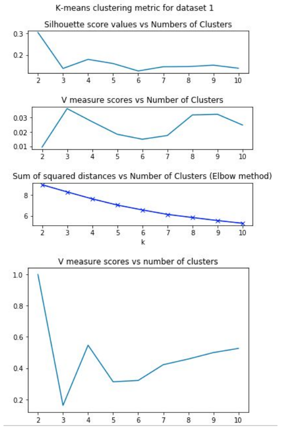
Part 3 - Dimensionality Reduction + Clustering

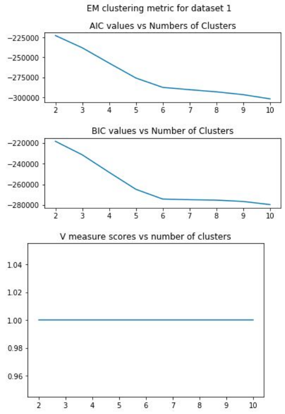
In this section, we will reproduce the clustering experiments after we reduce the dimensions of the original dataset.

Steel plates dataset -> PCA & ICA

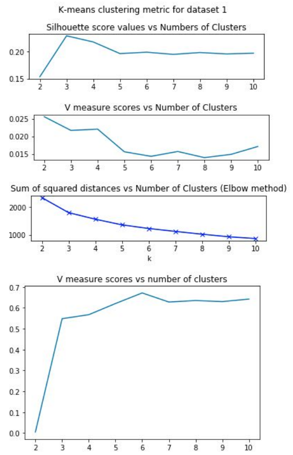


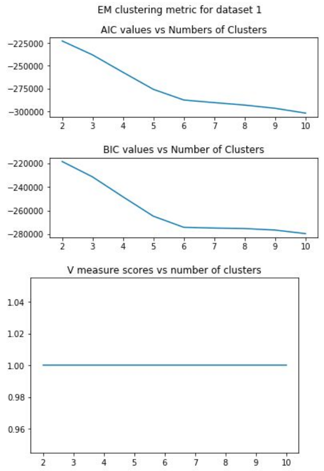


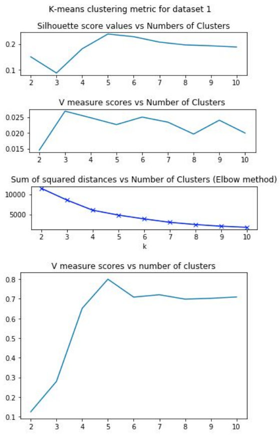


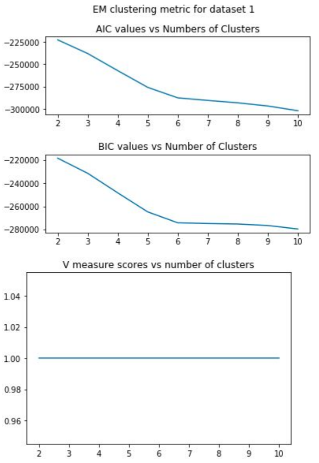


Steel plates dataset -> RP & Isomap

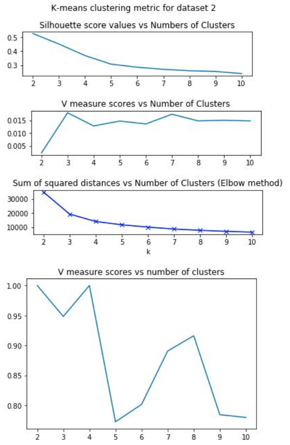


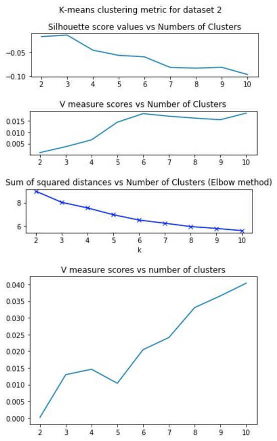


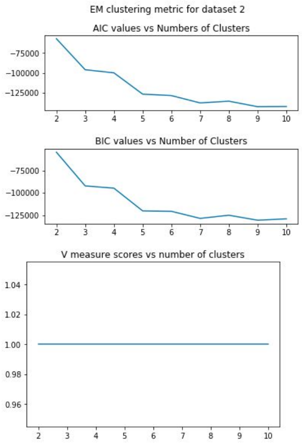




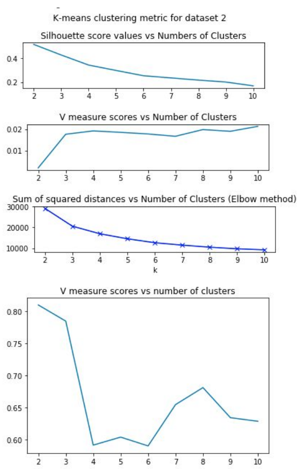
Contraceptives dataset -> PCA & ICA

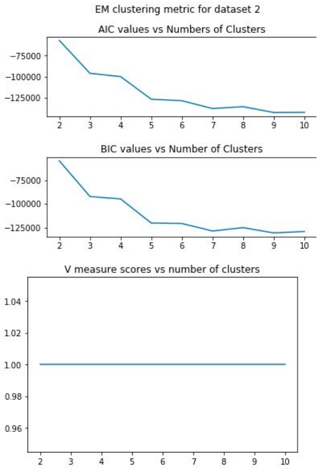


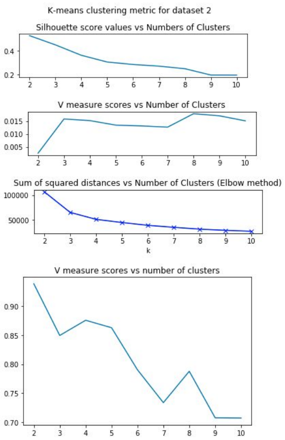


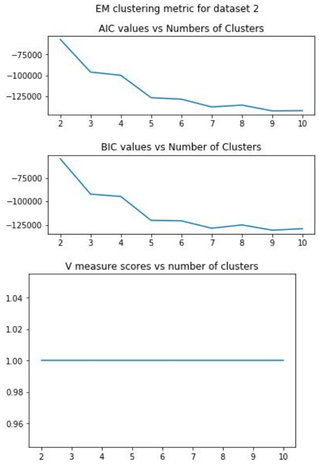


Contraceptives dataset -> RP & Isomap





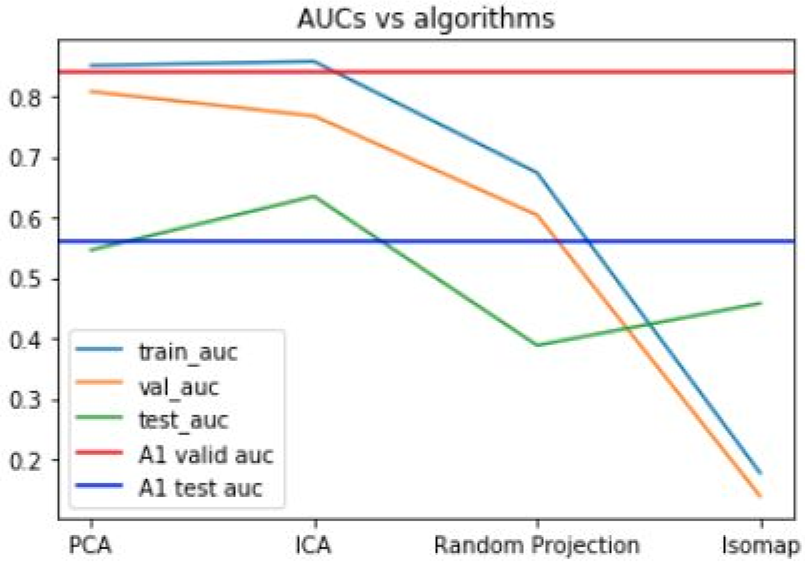


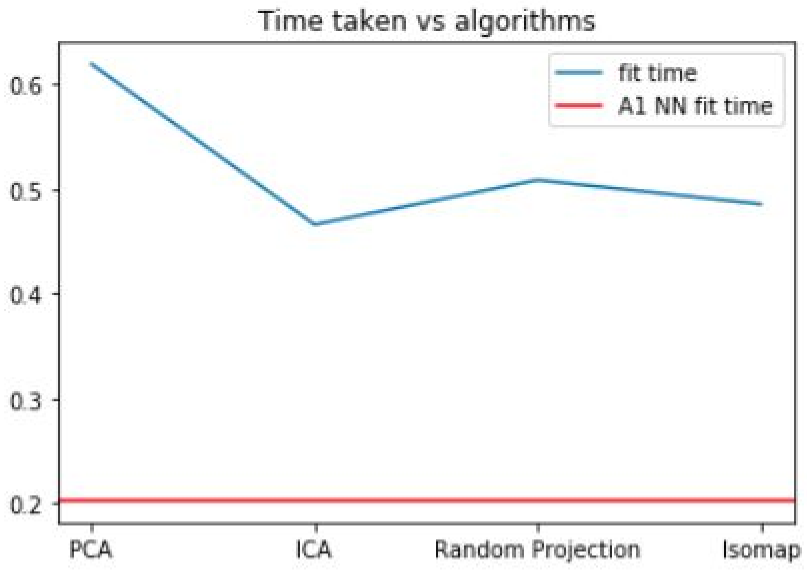


The above charts show the results of clustering algorithms after applying dimensionality reduction algorithms on both datasets. Applying K-Means and EM after PCA seems to generate charts with similar shapes for both datasets, possibly due to the conservative nature of choosing more dimensions to comprehensively explain the original variance. However, the v-measure of the original clusters vs the clusters after applying PCA seem to change as we increase the number of clusters. Also, applying PCA followed by K-Means improved the Silhouette score of the steel plates dataset. This is possibly because the post PCA features successfully captured the variance of the dataset. Interestingly, we also observe that the v-measure of clustering before and after applying dimensionality reduction algorithms seem to stay the same for EM. Perhaps the reason is because EM assigns a probability to each sample for any clusters, as opposed to hard assignment of labels in K-Means. Hence, the cluster labels do not fluctuate in the case of K-Means.

Part 4 - Dimensionality Reduction and Neural Network on Steel Plates dataset

In this section, we ran dimensionality reductions algorithms followed by a Neural Network (NN) on the Steel Plates dataset to analyse the impact of dimensionality reductions on performance and time complexity.



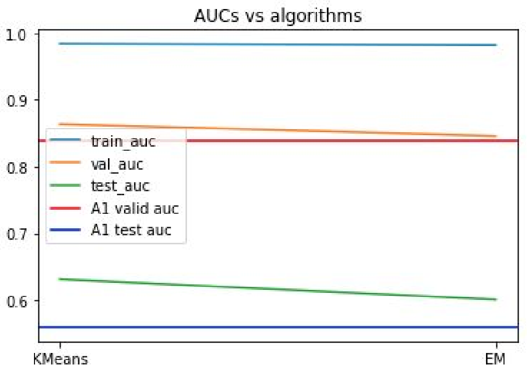


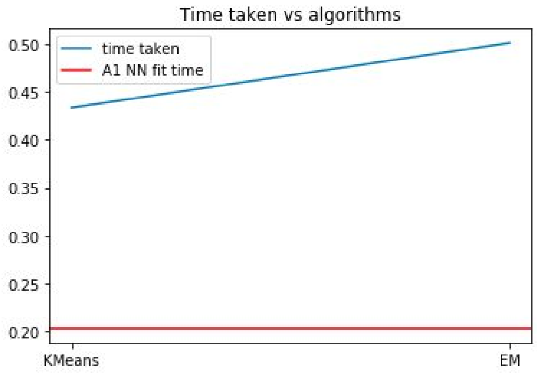
We compare the 4 Neural networks to the NN run on the Steel Plates dataset in assignment one and indicate two horizontal lines for the train and validation metrics. We reuse the same parameters for the NN. From the charts above, we observe that the assignment one validation and test AUCs are generally higher than that of any of the four NN after dimensionality reduction. The reason is because performing dimensionality reduction causes us to lose a proportion of the original information, resulting in lower AUCs.

Furthermore, we also conducted the time complexity analysis of the different NNs as compared to the NN in assignment one. It is interesting to note that the training time required for the four NNs is actually higher than that of the NN in assignment one. The reason is probably because the reduction in amount of information requires more iteration for the NNs to converge.

Part 5 - Clustering and Neural Network on Steel Plates dataset

In this section, we ran cluster algorithms followed by NN on the Steel Plates dataset to analyse the impact of adding clustering features on performance and time complexity.





Similar to part 4, we also compared the NNs trained in this assignment to the NN trained in assignment one. From the charts above, we can observe that the assignment one NN's validation and test AUC seems to be lower than what we generated for this assignment after K-Means and EM. This is probably because the additional feature from the cluster labels provides incremental information for the NN, resulting in better performance.

Furthermore, we also plotted out the time taken to fit the NN for assignment one and those in this section after applying K-Means and EM. The additional fit time required is not surprising considering that the dataset is the same, except that we include the cluster labels as an additional feature.

Conclusion

This assignment involved various dimensionality reduction and clustering analysis. The analysis really challenged my initial understanding of dimensionality reduction and clustering algorithms. In general, it is non-trivial to measure the impact of clustering algorithms. We chose the number of clusters based on various metrics. Furthermore, the dimensionality reduction algorithms are extremely useful to reduce the dimensions of high dimensional datasets, albeit not so much for visualization purposes as to resolve the issue of curse of dimensionality.

In summary, i think we should always conduct dimensionality reduction and clustering on our dataset to observe any performance / time complexity tradeoffs. If there are little performance sacrifices with huge time complexity gains, we should apply dimensionality reduction, unless the problem requires explainability of features to labels.

References

[1] Steel Plates Faults Data Set – UCI Machine Learning Repository
https://archive.ics.uci.edu/ml/datasets/Steel+Plates+Faults

[2] Contraceptive Method Choice Data Set - UCI Machine Learning Repository
https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice

[3] Scikit Learn: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html