CS7641 A3: Unsupervised Learning and Dimensionality Reduction

Trung Pham
tpham328@gatech.edu
https://www.overleaf.com/read/ddfypkpjjwrk#f3f858

I. Introduction

Unsupervised Learning techniques play a crucial role in uncovering hidden patterns and structures within data without relying on labeled outputs. This report explores the application of clustering and dimensionality reduction algorithms on two classification datasets that has been explored in project 1, supervised learning. By employing these algorithms, we want to gain insights into the intrinsic structure of the data and compare the performance with previously obtained supervised learning outcome using Neural Network learning algorithm.

The study of clustering and dimensionality reduction is particularly interesting because one allows us to identify natural groupings within data and the latter reduce its complexity, making it easier to visualize and analyze. Understand how these techniques can lead to more efficient data processing and potentially uncover relationships that are not immediately apparent through supervised learning alone.

Our selection of data from project 1 is particularly useful here because it has variety of features that we can draw connection from for clustering, as well as potentially noise that applying dimensionality reduction might help, although our results showed otherwise.

There are five main tasks in this project. The first is applying clustering algorithms (Expectation Maximization and K-Means) on the two dataset. Second, implementing dimensionality reduction (Principal Component Analysis, Independent Component Analysis and Randomized Projection) to transform the datasets. Third, reapplying clustering algorithms on the dimensionally reduced dataset, creating six combinations to evaluate which method performs best. Fourth, Re-running a neural network learner from previous project on the reduced dataset in step 2 to assess performance changes. Lastly, introducing clusters from step 1 as new features and evaluating the neural network's performance on this newly projected data. By systematically examining these approaches, we want to determine the most effective method for improving clustering performance, neural network efficiency, ultimately contribute to our machine learning and data analysis.

II. Hypothesis

Our hypothesises for this assignment are:

1. K-Means Clustering will outperformance Expectation Maximization

2. Principal Component Analysis (PCA) will provide the best improvement in clustering performance compared to Randomized Projection (RP) and Independent Component Analysis (ICA) and combining PCA with K-Means will yield the best clustering performance.

3. While dimensionality reduction may not improve the accuracy or F1 score of the neural network, it make the learner more efficient, requiring fewer epochs and less data to converge.

4. Using clusters from K-Means or EM as additional features will not significantly improve the neural network's performance.

III. Methodology and Experiments

This section outlines the methodologies employed in this project to apply clustering (EM and K-Means) and dimensionality reduction (PCA, ICA, RP) to the dataset and subsequently evaluate their performance.

K-Means is a clustering algorithm that partitions data into K number of clusters, where each data point belongs to the cluster of the nearest mean. It first initializes K centroids randomly, then assigns each data point to the nearest centroid, forming k clusters. It updates the centroids by calculating the new means from all data points in each clusters. It repeats the assignment and update steps until the centroids is no longer change significantly.

Expectation Maximization (EM) is used to find the most likely values of parametesr in probabilistic models, especially those with hidden or unobserved variables. In context of clustering, it is used to determine the best fit for the data. It first initialize the parameters (means, variances) for each cluster, then calcualte the probability that each data point belongs to each cluster based on current parameters. It update the parameters to maximize the likelihood of the data givent current cluster probabilities, and repeat calculation and update step until convergence.

Principal Component Analysis (PCA) is a linear dimensionality reduction technique that transforms the data into a new coordinate system where the greatest variance by any projection of the data comes to lie on the first few coordinate (principal components). Independent Component Analysis (ICA) is a technique that tranformed the data into additive, independent non-Gaussian components. It is used to reveal hidden factors that underlie sets of random variables. Randomized Projection (RP) is a technique that used random matrices to reduce the dimensionality of the data while preserve structure and distances between data points.

There are 5 main experiments being conducted. The first is applying clustering algorithm (KMeans and EM) to the two datasets. The second is applying dimensionality reduction algorithm (PCA, ICA, RP) to the two datasets. The third is applying clustering algorithm on top of the reduced dimension data in step 2. The fourth is using neural network learner from assignment 1 to re-run on the new reduced dimensionality dataset. The fifth is using same neural network learner on dataset where KMeans and EM clustering are used as new features. For experiment 1 to 3, performance is evaluated through silhouette score, comparison with true label or training time. For experiments 4 and 5, the performance is evaluated based on accuracy, F1 score as well as training time.

The two datasets we used are Diabetes and White wine quality dataset, which has been pre-processed from assignment 1. The details processing is described in assignment 1 report. Overall, Diabetes data is imbalanced. Target variable or label is binary classification (diabetes or no diabetes). There are 8 features and 768 records in the data. The white wine quality dataset is transformed to be balanced. Target variable or label are 3 classes classification (low, medium, high quality). There are 11 features and 4898 records in the data. Both datasets are used for experiments 1 to 3. Only white wine dataset is used for experiment 4 and 5.

IV. Clustering Algorithms Result and Analysis

The experiments first attempted to find the best number of cluster for each algorithm. Using the optimal cluster numbers, the data is explored using clustering algorithms. Their performance is compared using different metric like visual inspection, silhouette score, comparison with true label or training time.

A. K-Means Clustering Algorithm

From figure 1, using elbow method, the optimal K-means cluster is between 4 and 6. We also run a silhouette analysis to find that number of cluster equals 4 will give the best result because it gave highest silhouette score.

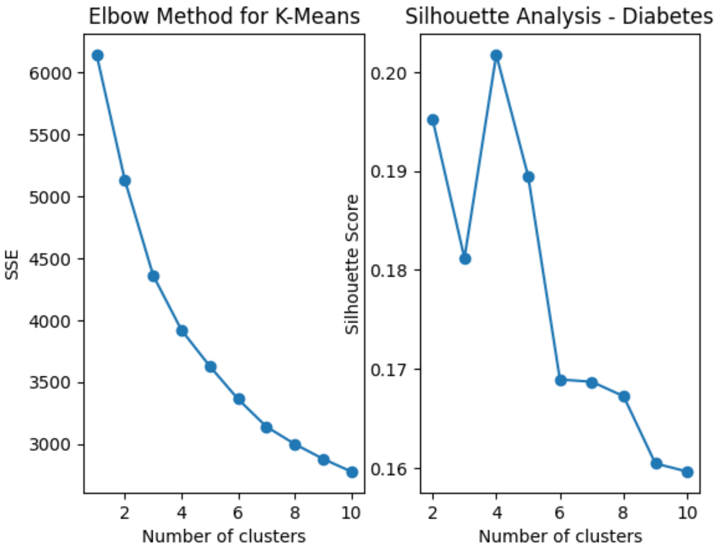

Fig. 1: Elbow method and silhouette analysis for Kmeans Cluster in Diabetes dataset

We applied K-Means clustering algorithm with 4 clusters to the Diabetes dataset. The visual inspection of the cluster in corresponding with Glucose and BMI value (two of the high correlating features with the true label) is not quite convincing (figure 7 - left plot). There is no clear distinction.

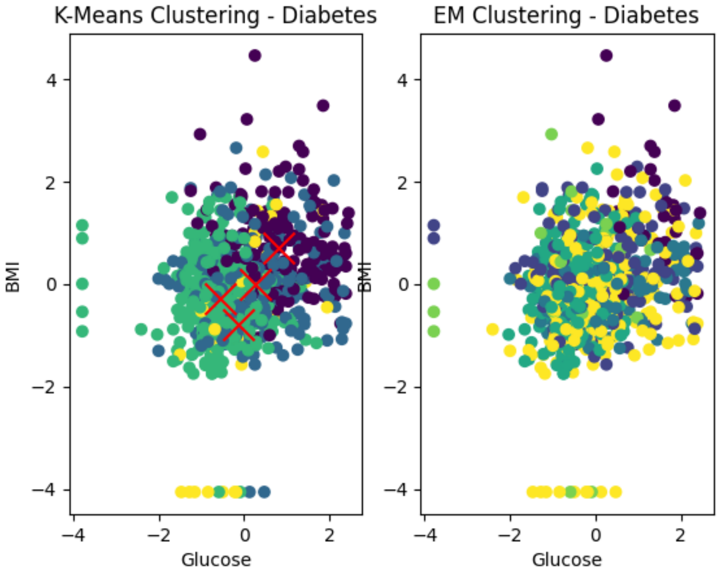

Fig. 2: K-Means and EM Clustering 2d view on Diabetes dataset

We also created Silhouette plot using the Silhouette score of the four cluster (figure3 - left plot). The red line in this plot represents the average silhouette score for all samples. Average Silhouette score around 0.2 indicates that the clusters are not well separated and there is some overlap. Orange and blue clusters have some points with negative score indicating their being assigned to wrong clusters. Overall, K-Means cluster might be reasonable but there could be improvements.

Similarly, we used elbow method and Silhouette analysis to find optimal K-Means cluster for Wine data (figure 4). We want to pick the elbow points where the SSE start decreasing slowly, but we also want to pick cluster numbers that resulted in high silhouette score. 2 cluster gave highest Silhouette score but also highest SSE which we do not want. We choose 6 clusters where Silhouette score is 0.14 and it is near the elbow in SSE plot.

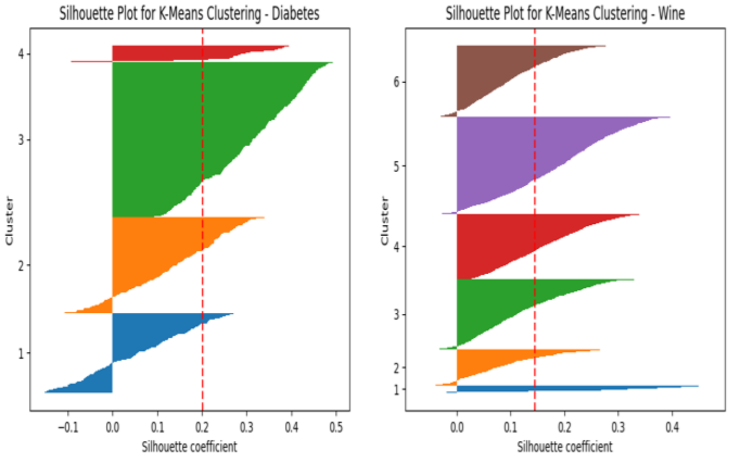

Fig. 3: Silhouette score for K-Means Cluster with different dataset

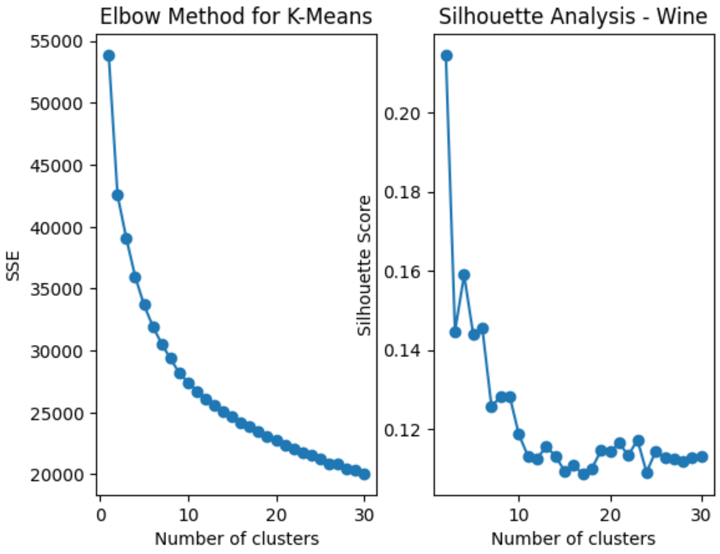

Fig. 4: Elbow method and silhouette analysis for Kmeans Cluster in Wine quality dataset

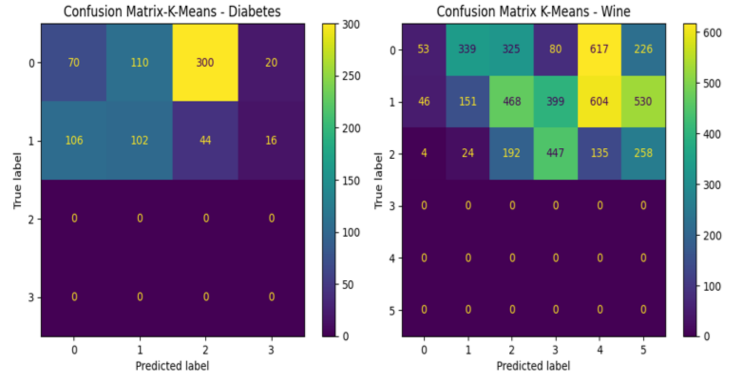

Fig. 5: Confusion Matrix - KMeans - Diabetes and Wine dataset

Visually, when plotting 6 clusters in a 2D graph with each axis represents Alcohol and density, two of the highest influence of wine quality, we saw no clear cluster patterns. It is understandable because 6 clusters are hard to view in two dimensions as they can overlap. We also inspected other features pairs but found no distinction in cluster pattern. However, when producing Silhouette plot, the result is promising. All six clusters achieved higher score than the average line. We have some negative Silhouette score representing wrong clustering assignment but they are small, much smaller than in Diabetes experiment (figure 3 - right plot). The reason we did not see clear cluster in 2d plot is because of high number of clusters might overlap in 2 dimension space or because of noise and outlier in the data.

Figure 5 is the confusion matrix table with y-axis is the true label and x-axis is the predicted cluster. For Diabetes data, if we used the clustering to predict the true value, only cluster with predicted label 2 have majority 'no diabites' label, the rest of clusters are 50/50 random guess in term of prediction for diabetes. The same things can be said for Wine dataset, there is no clear pattern or prediction of wine quality using KMeans cluster.

B. Expectation Maximization

Additionally, we applied EM cluster to the two datasets. To find the optimal number of clusters, we calculate BIC and AIC value using different number of clusters and choose cluster numbers that can minimize AIC BIC. In figure 6, for Diabetes, the elbow for BIC blue line starts around 6, and for Wine dataset, the elbow starts around 4. We will use 6 EM Cluster for Diabetes and 4 EM Cluster for Wine data.

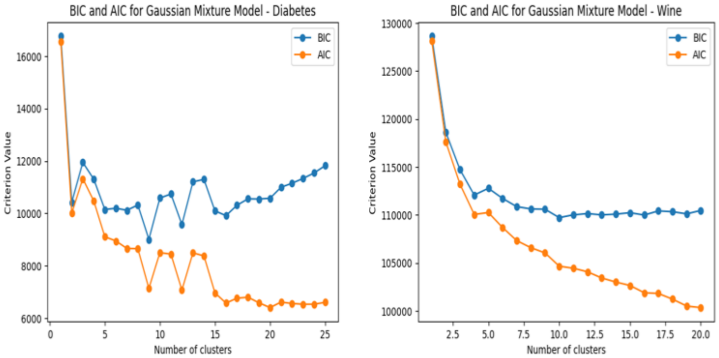

Fig. 6: BIC and AIC line to find optimal EM cluster number

Looking at EM cluster in 2d view with 2 high influence features on both dataset (fig 2. and fig 3. right plot), we can hardly see any cluster pattern. The cluster on Wine dataset look a bit better than wine. In term of visual, EM cluster does not look as good as K-Means.

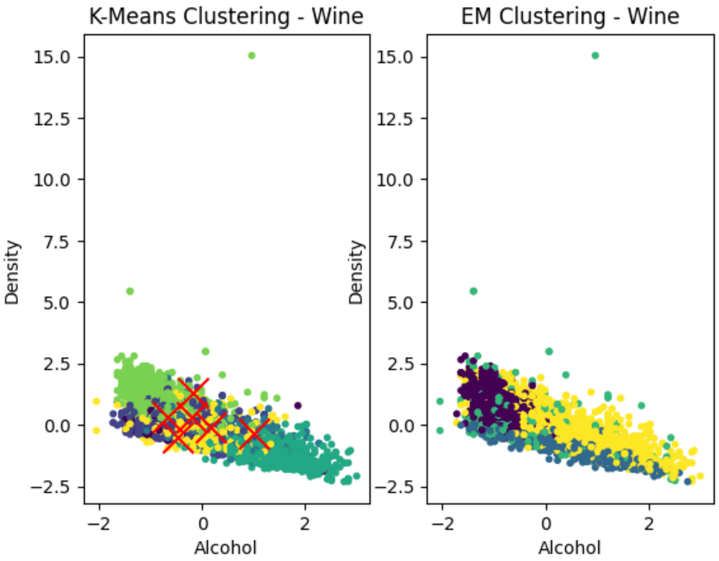

Fig. 7: K-Means and EM Clustering 2d view on Wine dataset

Next, Silhouette score of EM Cluster is worse than K-Means cluster. The plot on both dataset (fig. 8) show many negative score indicating many bad assignments of cluster. The distribution of points are uneven and average score is lower than K-Means cluster's score. At least in K-Means cluster, Silhouette score showed potential of clustering. Silhouette score of EM cluster indicates bad clustering. Similarly to KMeans cluster, when comparing with the true label, EM Cluster showed no prediction potential for both diabetes and wine quality.

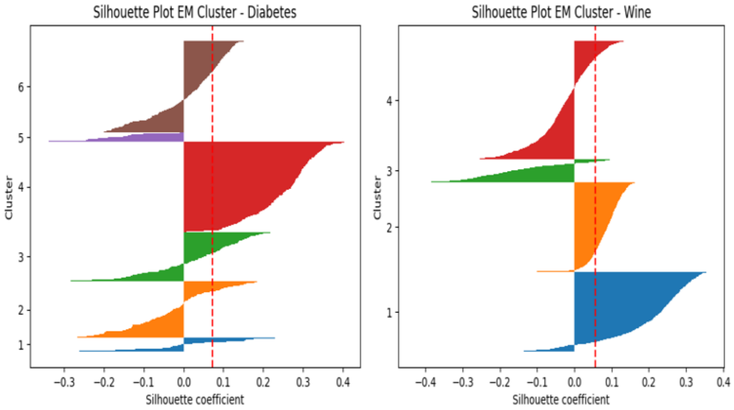

Fig. 8: Silhouette score for EM Cluster with different dataset

The last criteria being compared is training time between K-Means and EM algorithm. With small number of clusters, EM Cluster required similar or less computational power compared to K-Means. As the number of clusters increases, the time it took to train EM Cluster increase steeply compared to K-means which barely increases. This indicates that K-Means tend to converge quickly and more scalable due to K-Means directly assigns points to cluster and update centroids. EM involves calculating probabilities and update distribution which is computationally intense. EM algorithm might become impractical for large number of clusters.

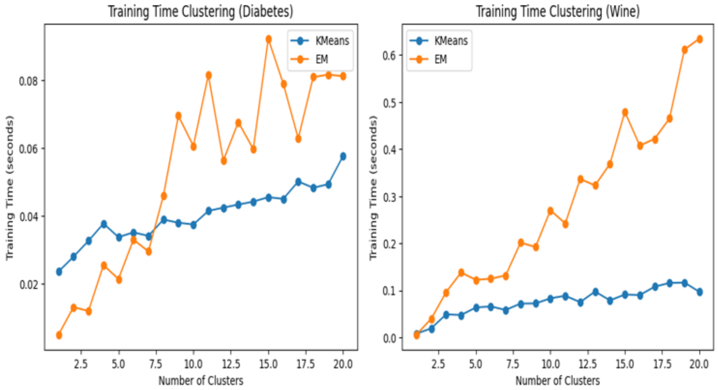

Fig. 9: Compare training time of KMeans and EM Clustering

In summary, K-Means clusters are computational efficient, handle large datasets better and separate data into cluster better than EM clusters. Training time of K-Means are better than EM due to simpler algorithm. KMeans Silhouette score is also better, indicating better clustering performance, the algorithm made less mistake when assigning points to correct clusters.

V. Dimensionality Reduction Analysis and Combination with Clustering

In this section, we combine experiment in step 2 and 3 by first exploring the dimension reduction analysis then applying clustering algorithm in these new transformed data. Similar to step 1, we will find the optimal hyperparameters before running the algorithm. For clustering, we tuned the number of clusters, and for dimension reduction, we tuned the number of components.

A. Principal Component Analysis

For PCA, the distribution of Eigenvalues indicates the amount of variance captured by each principal components, with larger eigenvalues representing components that capture more variance and smaller eigenvalues indicating smaller captured variance. Plotting eigenvalues as cummulative explained variance help us pick numbers of principal components we want.

For Diabetes dataset, 6 Principal Components explained 90% of variance in the data, while in Wine dataset (fig. 10 top left), 8 Principal Components explained 92% of variance in the data (fig. 12 top left). Applying PCA, we reduced 9 features to 6 PCs in Diabetes data and 11 features to 8 PCs in Wine data.

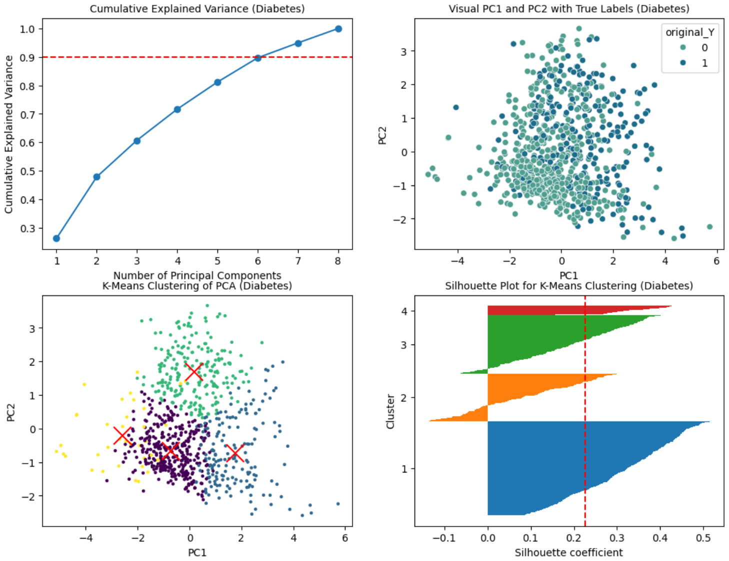

Fig. 10: Top left and right is PCA on Diabetes dataset, bottom left and right is KMeans Clustering applied to PCA Diabetes dataset

Visually, when plotting the true label against PC1 and PC2, which capture the most significant variation in the data, we could not detect any clear distinction in both dataset (Fig. 10 and 12, top right). Compared with true label, there is also no clear pattern. The reason is limited variance captured by PC1 and PC2, even though they are important, together they only capture 50% of variation in both data. Both dataset are complicated and need more components. In addition, some classes might be inherently overlapping in limited 2D space. Also, some features might not be relevant, PCA reduces dimensions based on variance but not help on class separability.

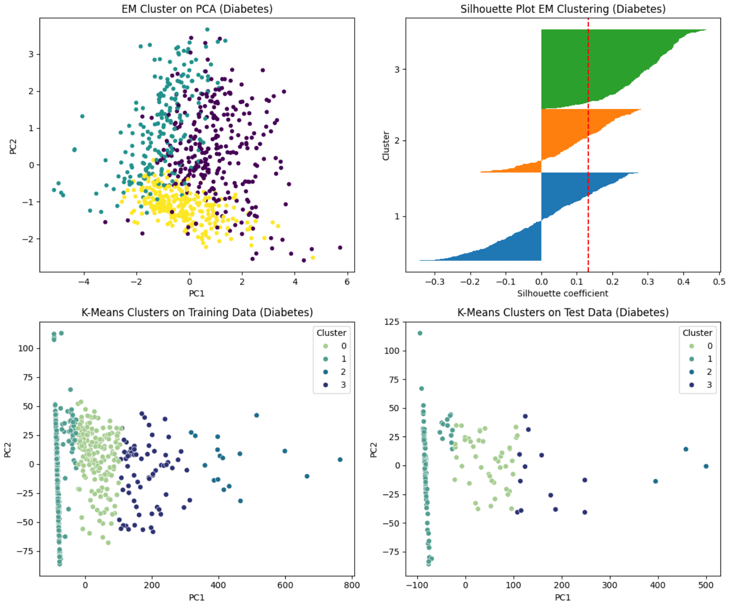

Fig. 11: Top plots are EM Cluster performance on PCA, lower plots are Kmeans Cluster learner performance on Diabetes dataset

If PCA alone does not help, applying KMeans clustering show significant improvement in cluster performance. Fig. 10 and Fig. 12 lower left plots show clear cluster even though they are not completely separated. This is because PCA reduces noise and transform feature space such that clusters may become more spherical and equally sized, which suits KMeans algorithm. High dimensional data often suffers from the curse of dimensionality, where distance measures become less meaningful. PCA reduces the dimensionality, making clustering algorithm more effective.

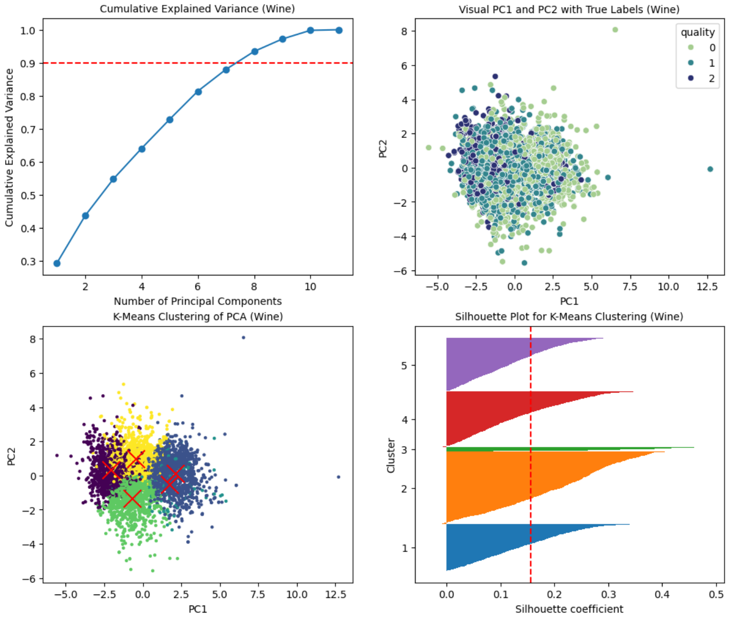

Fig. 12: Top left and right is PCA on Wine dataset, bottom left and right is KMeans Clustering applied on top PCA Wine dataset

In term of Silhouette score, KMeans on PCA transformed of both datasets provide good result. The average scores are higher than the score of K-Means without PCA, indicating improvement. Especially in wine quality, all clusters Silhouette plots look well separated, equally distributed and little mis-assignment.

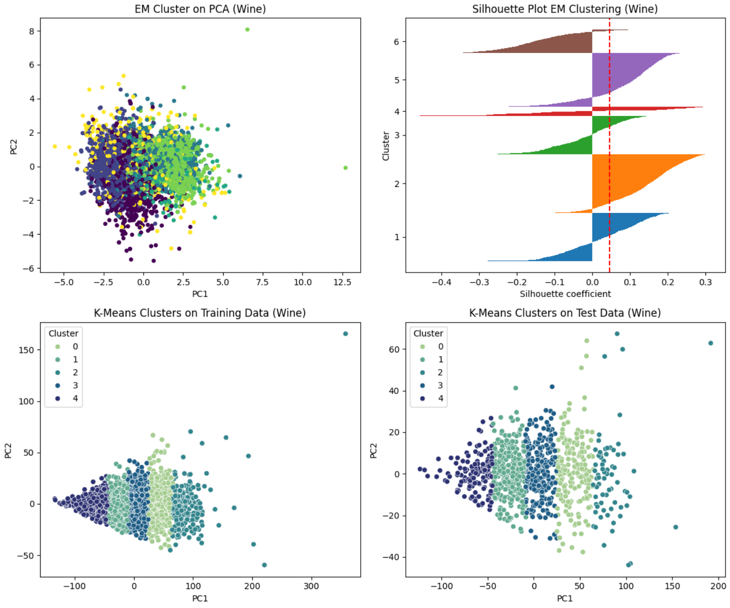

Fig. 13: Top plots are EM Cluster performance on PCA, lower plots are Kmeans Cluster learner performance on Wine dataset

In addition to K-Means, when applying EM clustering algorithm to PCA transformed data. There is little to no improvement in wine dataset both visually and in Sihoulette score (fig. 13). In diabetes dataset, the application shows some improvements with higher average Silhouette score. The EM cluster looks visually separated (fig. 11 top left) but sometimes points are still sorted into wrong cluster (fig. 11 top right-blue and orange). In general, EM Cluster does not perform as good as K-Means cluster in PCA dataset, which reinforce hypothesis 1.

Given the good result of KMeans on PCA, we attempt to train a cluster model using the two datasets. The results from both dataset in both training and testing set show very good result (Fig.11 and fig.

13 bottom). The cluster using on top of PCA data are quite defined, separated and litte mis-assigned. It might not have predicting power on our target label but potentially the clusters indicate unknown labels we do not know.

B. Independent Component Analysis

In ICA, the distribution of independent components are typically highly kurtotic, meaning heavier tails and sharper peaks. To find the numbers of components used in Independent Component Analysis (ICA) we choose based on the elbow in average kurtosis value line. As high kurtosis indicate significant independent components. We run all possible number of components in ICA, calculate each component kurtosis to calculate the average. For diabetes dataset, we transformed the data into 5 ICs and for wine dataset, we transformed the data into 7 ICs (fig 14 -16 top left plot is kurtosis line).

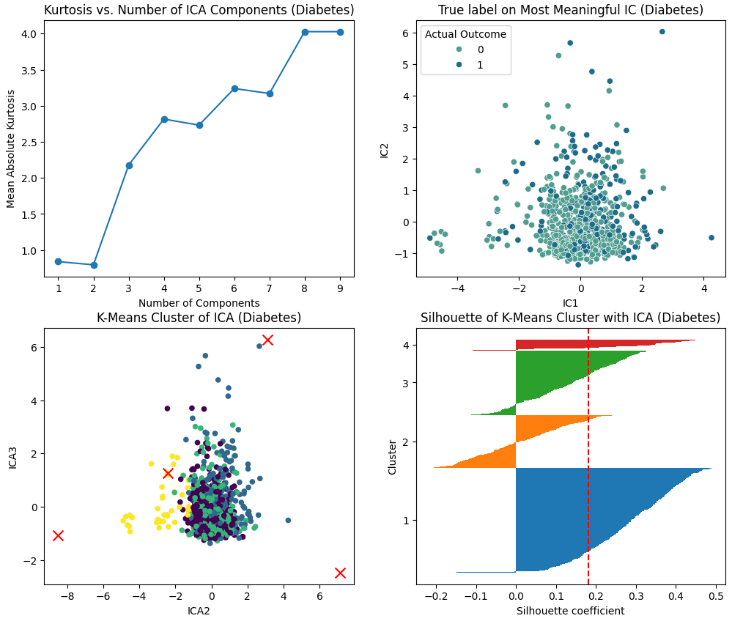

Fig. 14: Top left and right is ICA on Diabetes dataset, bottom left and right is KMeans Clustering applied to ICA Diabetes dataset

ICA is a technique that separating features into additive, independent non-Gaussian components. Kurtosis is used to identify non-Gaussianity of the components. ICA seeks to maximize or minimize the kurtosis of the components. Non-Gaussian components will have absolute kurtosis values higher than 3. For diabetes data with 5 ICs, there are 3 significant non-Gaussian components. For wine data with 7 components, there are 4 significant non-Gaussian components.

When comparing with true label visually through 2d plot (fig. 14 and 16 top right) or empirically through confusion matrix, there seem to be no connection or patterns, indicating no predicting power.

Applying K-Means cluster to ICA data, visually there is no clear cluster pattern or separation on Diabetes data but cluster looks good on wine dataset (fig 14 and fig 16 bottom left). The Silhouette plot confirmed this observation. Diabetes data have many mis-assigned points, low average and unequal distribution. The Silhouette on wine data however look acceptable with clear cluster pattern with one cluster have less distribution. Overall, KMeans cluster algorithm performent is acceptable in wine dataset but no Diabetes dataset. Both performance are worse than KMeans cluster on PCA data.

In addition, applying EM Cluster on both ICA data do not produce good result, which is confirmed visually in 2d plot and Silhouette plot (fig 15 top and fig. 17 top). There are many points that are mis-assigned to further clusters. Overall, performance is not as good as KMeans with PCA.

Because Kmeans cluster algorithm on ICA have acceptable result, we attempt to build a cluster model using Kmeans algorithm with ICA wine dataset. The result again look acceptable, with clear cluster color but not clear separation (fig. 17 bottom). We also built a confusion matrix to compare this cluster with actual label wine quality. Overall, only 1 cluster have potential to predict ’medium’ quality, the other 2 clusters are inconclusive when predicting the other classes (’high’ and ’low’). There is little to no potential to predict wine quality using cluster in ICA.

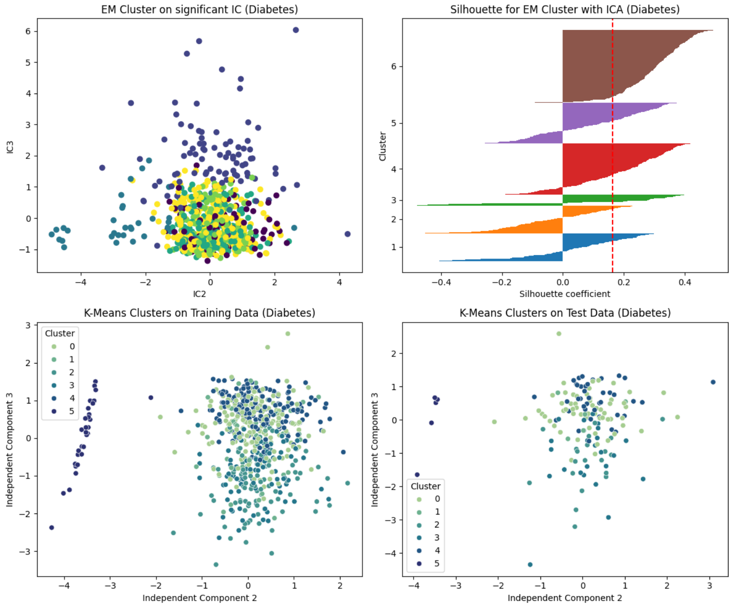

Fig. 15: Top plots are EM Cluster performance on ICA, lower plots are Kmeans Cluster learner performance on Diabetes dataset

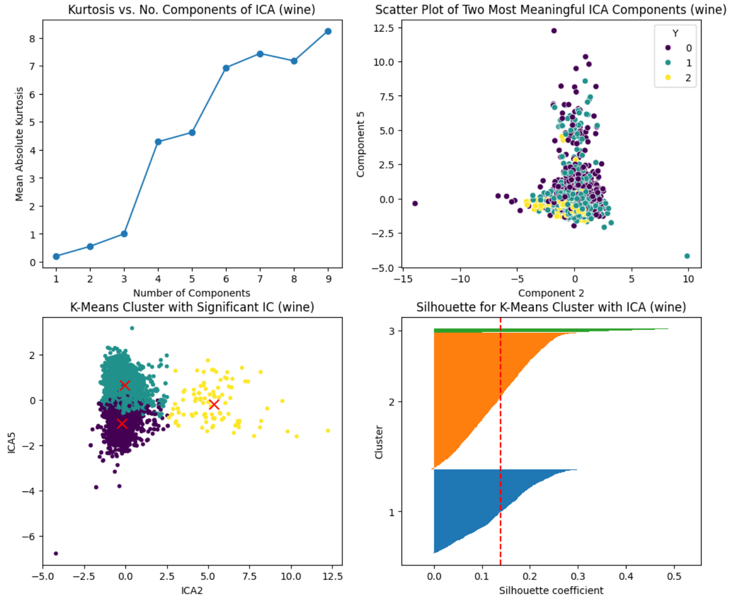

Fig. 16: Top left and right is ICA on Wine dataset, bottom left and right is KMeans Clustering applied on top ICA Wine dataset

1) Randomized Projections

We used Reconstruction Error to find the numbers of components for Randomized Projection. The error line however showed no clear elbow and inconclusive. Since we want to reduce dimension of data, we pick number 4 because we want to reduce the number of features in half. The reconstruction error of 0.523 and 0.615 on diabetes and wine dataset respectively are acceptable. The Randomized Projection transformed the data into new subspace of 4 dimensions using random matrix.

Similar to other dimension reduction technique, when comparing with true label. There is no clear pattern or connection to the target label (fig 18 and fig.19 top right plot). All the colors are mixed up indicating overlapping and mis-assignment.

Applying EM clustering on top of RP transformed data show no improvement as all cluster colors are still mixed up with a

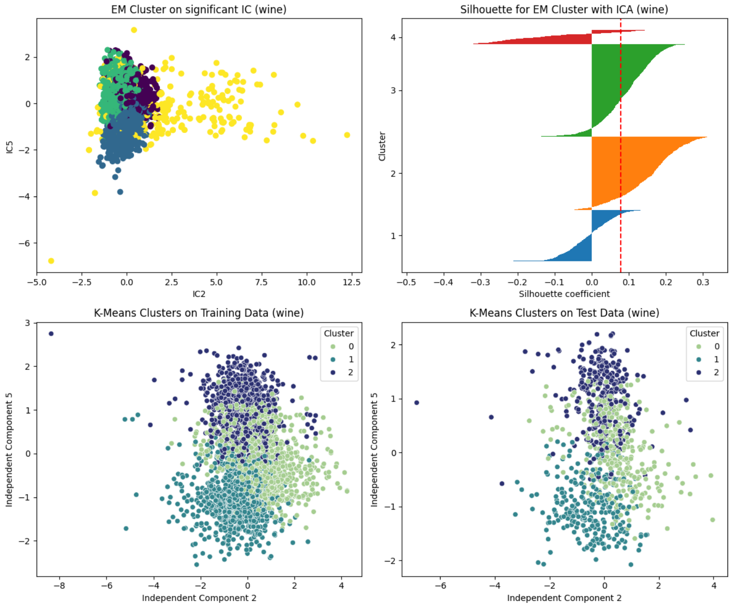

Fig. 17: Top plots are EM Cluster performance on ICA, lower plots are Kmeans Cluster learner performance on Wine dataset

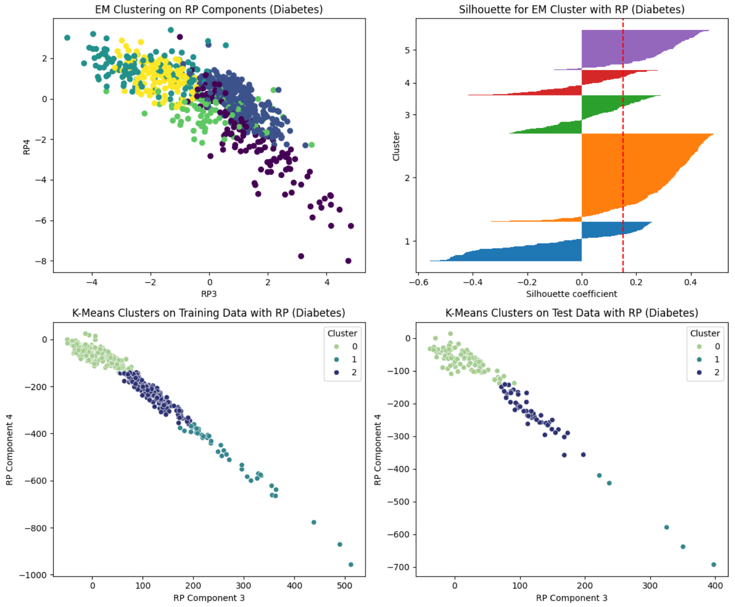

Fig. 19: Top plots are EM Cluster performance on RP, lower plots are Kmeans Cluster learner performance on RP Diabetes dataset

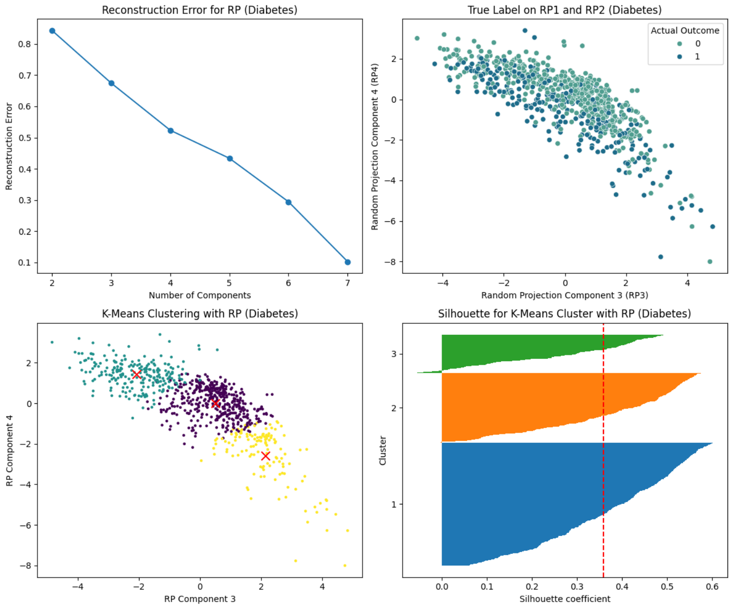

Fig. 18: Top left and right is RP on Diabetes dataset, bottom left and right is KMeans Clustering applied to RP Diabetes dataset

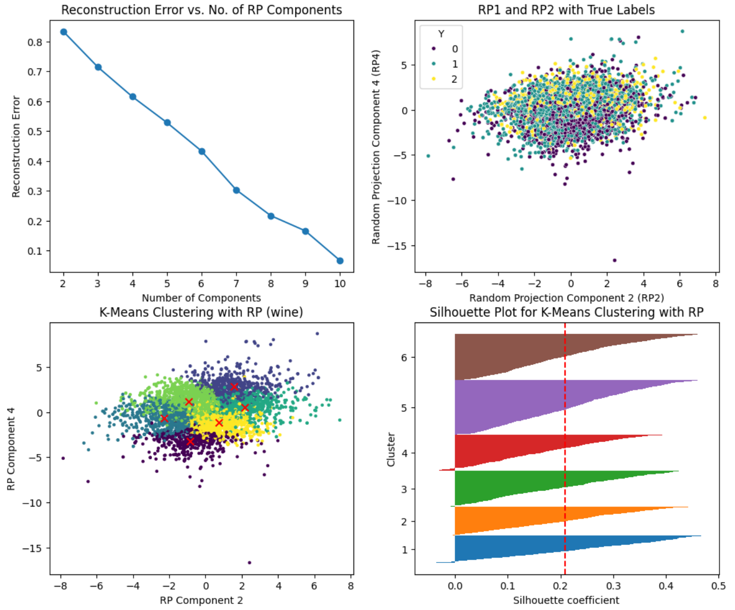

Fig. 20: Top left and right is RP on Wine dataset, bottom left and right is KMeans Clustering applied on top RP Wine dataset

lot of negative Silhouette score. However, when applying K-means algorithm on top of RP transformed data, the results are surprisingly good (fig. 18 and fig. 20 lower plots). We can see separation of clusters, which is confirmed by Silhouette score with little mis-assignment. There are equal distribution of points in each cluster in both Diabetes dataset.

Followed this discover, we built a K-Means cluster learner using the two datasets. The results are good, showing in lower plots in fig. 19 and fig. 21. When comparing with true label, however, there is no connection to our target label. Part of the reason is because diabetes data is imbalanced. The wine quality data is balanced so there is potential to use cluster for prediction but the accuracy is low, only around 60%. The cluster might be an intrinsic feature that we do not know about.

There are couple reason why K-Means cluster on RP performed good. First, Randomized Projection tends to preserve pairwise distances between data points. This perservation aligns well with K-Means algorithm, which relies on distance. RP reduces dimensionality, which essentially reduce noise or complexity in the data, which boost K-Means performance. RP also maintained underlying linear structure of the data, which help K-Means algorithm.

In all experiments, none of dimension reduction can help predict our target label. PCA is the one that is simplest, fastest, and easiest to implement. Using K-Means cluster on PCA produce the best cluster performance. Surprisingly, K-Means cluster on RP data produce similarly good result, opening up further opportunities for study.

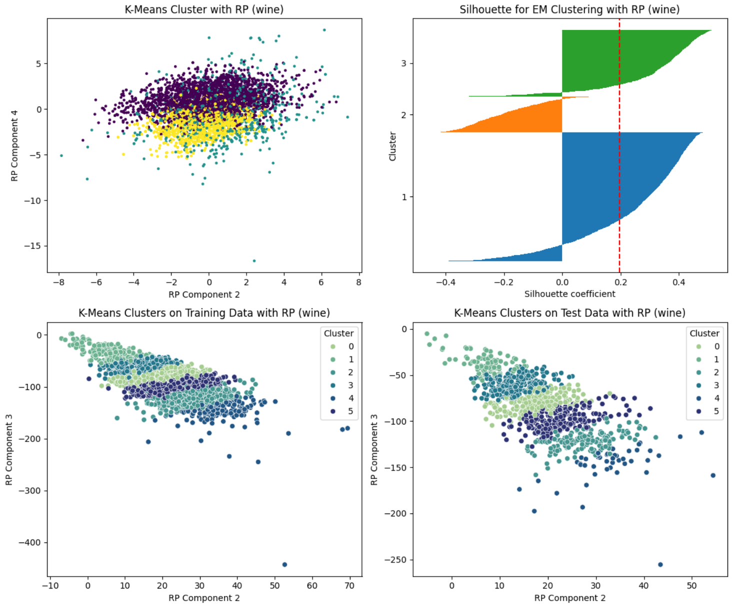

Fig. 21: Top plots are EM Cluster performance on ICA, lower plots are Kmeans Cluster learner performance on RP Wine dataset

VI. Neural Network Learning on Dimensionality Reduction dataset

We transformed the wine dataset using dimensionality reduction techniques. Because inputs changed, we re-tune our hyperparameters from project 1 using optuna package for each transformed dataset. Using good combination of learning rate, layers and hidden units, we evaluate the new neural network performance on new dataset compared to original result from project 1.

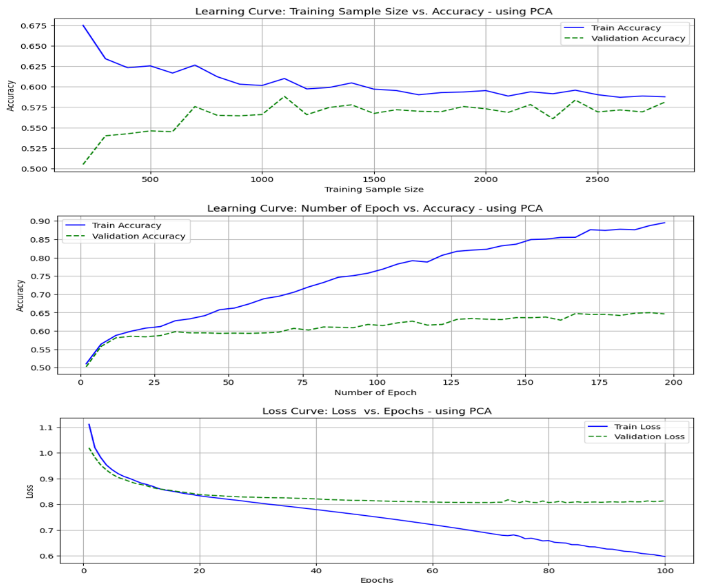

Fig. 22: Neural Network on PCA transformed data

Using the learning curve against training sample size, our neural network (NN) converge faster than the original. Supporting the curse of dimensionality theory, using dimensionality reduction technique help reduce the number of features, making the learner require less data to converge. RP model converge earliest followed by PCA and ICA, due to RP transformed data have the least features.

In term of iteration, performing NN on transformed dataset required less epochs (or number of iteration) compared to the original (epoch = 30). The new transformed data sometime prompt early stop on epoch below 20 before overfitting occurs. Loss curve on all three transformed dataset also support lower number of epochs before overfitting. The learning curve or validation curve produce very similar patterns to original dataset, indicating that the transformed dataset manage to capture most of the important features required to train neural network.

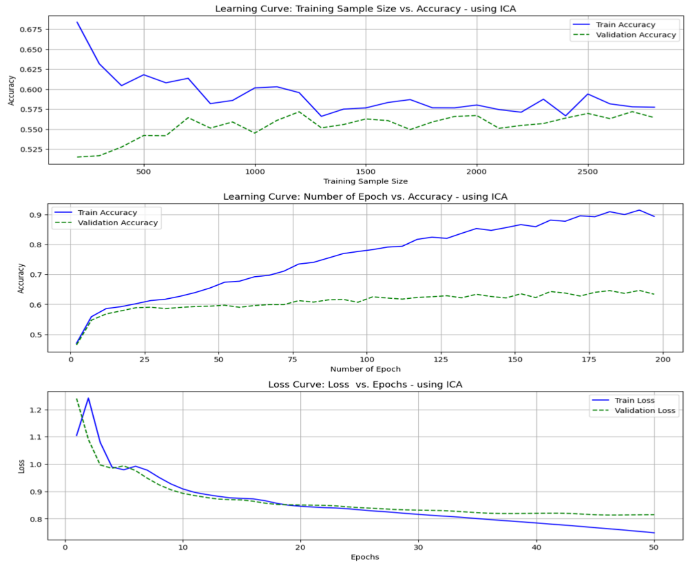

Fig. 23: Neural Network on ICA transformed data

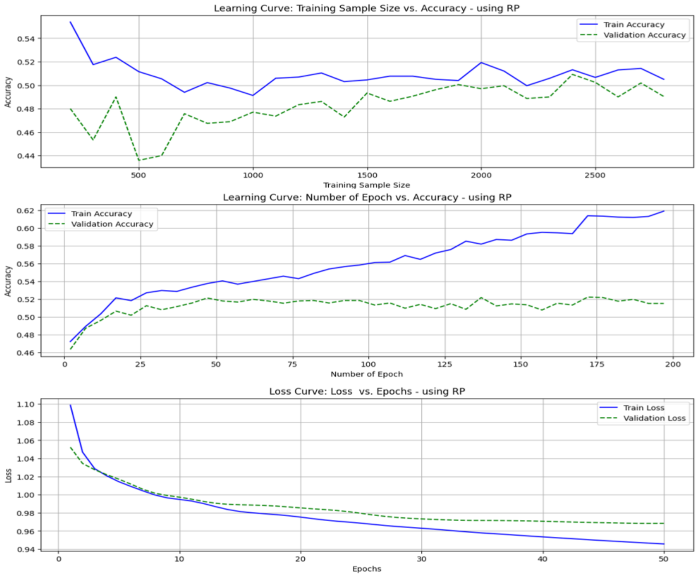

Fig. 24: Neural Network on RP transformed data

In term of training time, the original data takes longest time followed by ICA, PCA and RP data. This is as expected following the same order of dimensionality of each dataset.

Class	PCA	ICA	RP	Original
low	65%	61%	49%	68%
medium	62%	62%	60%	64%
high	51%	47%	13%	51%
accuracy	61%	59%	51%	63%

TABLE I: Neural Network learner on dimensionality reduced data

In term of quantitative results (table 1), all three dataset produce good result with RP performs worst and PCA performance best. However, they are all worse when comparing with original result. RP dataset performs bad when predicting 'high' class due to imbalance data, as well as possibility of removing importance information when reducing dimension. PCA dataset produce results very close to original data, reinforcing our hypothesis about the effectiveness of PCA. Overall, transforming the data in PCA can achieve close performance of original data while making the learner more efficient.

VII. Neural Network Learning on dataset with Clustering as new feature

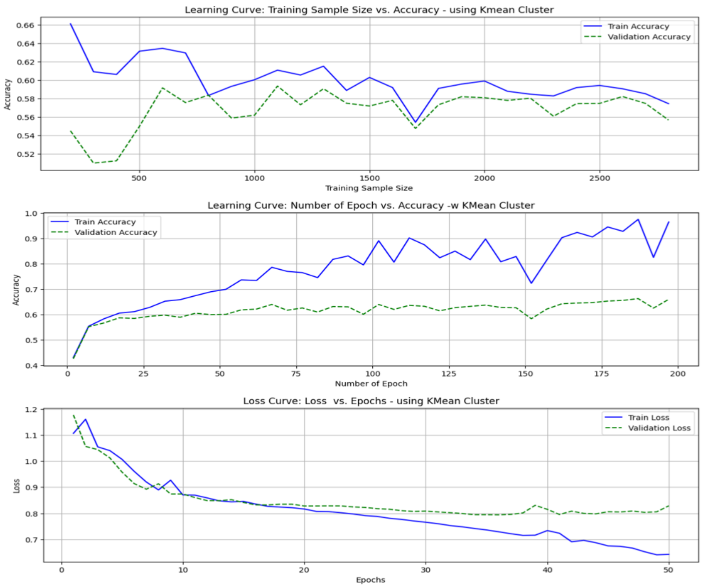

Fig. 25: Neural Network on data with KMeans Cluster

Adding additional clustering features produce similar learning curve with the original data. The curve however indicating higher accuracy given smaller training sample size. This is because new features have been able to capture certain intrinsic information of the data.

When plotting learning curve against iteration number epoch, the curve prompt early stop around 25, which is a bit lower than the original's of 30, implicating that new features have capture important information of the original data. In addition, the training loss and validation loss curve also diverge earlier at smaller epoch to support this idea again. Using clustering feature, the learner can generalize good with lower number of iteration compared to original.

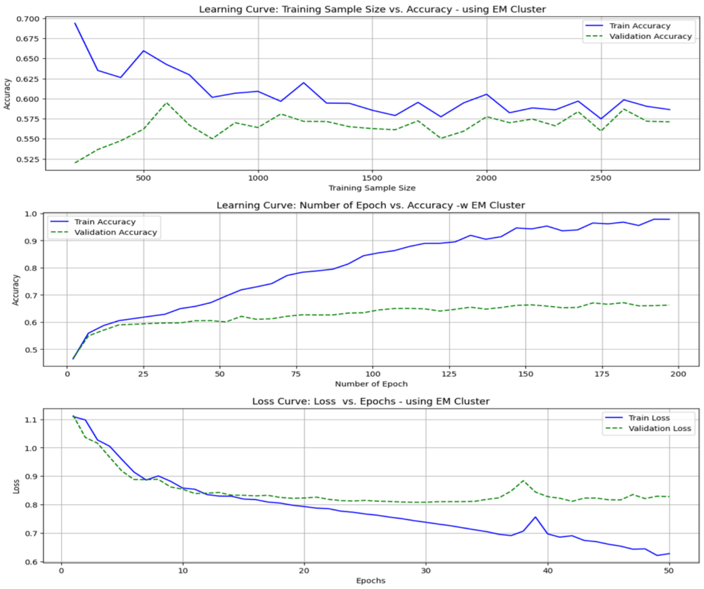

Fig. 26: Neural Network on data with EM Cluster

In term of accuracy on validation curve, there is no improvement using additional feature. The curve ultimate accuracy is about 60%, which is little lower than the original. Higher epoch can increase accuracy but also increase overfitting as indicated by the widening gap between training and validation curves, which is not ideal. This supports the Bias-Variance trade-off theory. Overall, adding new features data does not boost accuracy without sacrificing generalization.

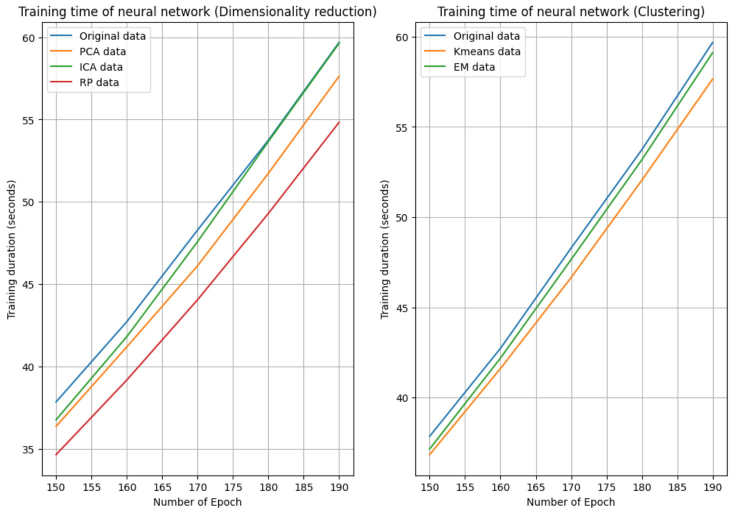

Fig. 27: Neural Network training time

Again, in term of quantitative result using F1 score and accuracy, adding KMeans and EM cluster does not improve performance (Table 2). Because adding clusters as additional features may introduce noise or redundant information, disrupting the network's ability to learn from original features. The added complexity can lead to overfitting, ultimately reducing accuracy and performance compared to using the original data alone.

Class	KMeans	EM	Original
low	64%	62%	68%
medium	62%	58%	64%
high	46%	62%	51%
accuracy	60%	60%	63%

TABLE II: Neural Network learner on data featured cluster

VIII. Limitation and Improvement

The choice of dimensionality reduction techniques like PCA, ICA and RP can significantly impact results, with each method having its own assumptions and limitations. While K-Means scales well with large dataset, EM can be computationally intensive, limit its effectiveness for large scale analysis. Clusters produced by transformed data may lack interpretability, challenging insights and practical application. We saw that cluster is intrinsic feature that potentially can be explored. Findings in this report are specific to the dataset used, so different dataset may yield varying results, limiting the generalization ability. Notably, K-Means clustering on RP transformed data shows promising performance similar to PCA, presenting opportunity for further research.

IX. Conclusion

Our hypothesis for this assignment were largely supported by the findings. K-Means Cluster outperformed Expectation Maximization in terms of clustering and training time. Principal Components Analysis provided the best improvement in clustering performance, combining with K-Means yielded the most effective results. While dimensionality reduction did not significantly enhance the accuracy or F1 score of the neural network, it did improve efficiency by reducing epochs and data required to converge. Additionally, using clusters as additional features did not significantly enhance the neural network performance probabliy due to our specific datasets. Lastly, there are potential of K-Means on Randomized Projection dataset effectiveness, offering opportunity for further research.

X. Resources

[1] *API Reference.* Scikit-learn. https://scikit-learn.org.