Unsupervised Learning and Dimensionality Reduction

CS7641: Machine Learning

Kim, Irene
skim3364@gatech.edu

I. Objective

The objective of this report is to compare and contrast the unsupervised learning algorithms. Specifically, two clustering algorithms and four dimensionality reduction algorithms are explored. For clustering algorithms, k-means clustering and expected maximization(EM) are used. The selected dimensionality reduction algorithms are principal component analysis(PCA), independent component analysis(ICA), randomized projections(RP), and univariate feature selection(UFS). The remainder of the paper is structured as follows: Section 2 provides a brief explanation of chosen datasets. Section 3 explores the result of applying clustering algorithms to the datasets. Section 4 describes how algorithms perform on the same datasets using dimensionality reduction. Section 5 then reruns the clustering algorithm on the dataset that dimensionality reduction was performed. Total of 16 combinations of datasets will be discussed. Section 6 applies the neural network on newly projected data by dimensionality reduction algorithm. Section 7 experiments with the neural network on the new dataset composed of features that are result of clustering. Finally, the work is summarized in section 8. The code is released on GitHub [1].

II. Dataset

The selected dataset for this assignment is *Wisconsin Breast Cancer* dataset(WBC) and *Pima Indians Diabetes* dataset (PID). These are the same dataset used in all of the previous assignments. For the same reason as previous assignments, these datasets are chosen because they deal with real-world problems such as health. Handling real-world problem with machine learning is interesting as they illuminate the great potential of machine learning to help human and the community. Secondly, the structural aspect of these datasets is interesting. They both are imbalanced datasets with similar dataset sizes. The only difference is the number of features. The WBS dataset contains 30 features whereas the PID dataset only contains 8 attributes excluding the label column. In the previous assignments, this difference led to differing results. A further description of the datasets is in Table 1.

[1]The code is available at GitHub: https://github.com/skirenekim/CS7641-Machine-Learning/tree/main/HW3

TABLE I
Dataset Details

Data	Instances	Positive Class	Negative Class
WBS [1]	569	357	212
PID [2]	768	500	268

III. Clustering

In this section, two clustering algorithms, K-means and EM algorithms, are performed on each dataset. Clustering algorithms learn problems in an unsupervised fashion. Unlike supervised learning, the clustering algorithm focuses on finding a more compact way of describing data rather than finding an optimal function approximation.

A. *K-means*

The K-means algorithm is a clustering algorithm that first picks k centers. Each selected centers claims its closest points and recomputes the centers by averaging the clustered points. It repeats this process until convergence. Computing the closest comes down to selecting the distance. In this experiment, Euclidean distance is used.

Why Euclidean Distance?

Euclidean distance is chosen because the K-means algorithm is based on pairwise Euclidean distances between points and repeatedly assigning data points to the closest centroid. In particular, the partition of point x is going to be the minimum over all the clusters of the distance between x and the center of that cluster. This sum of squared deviations from the center of that cluster is equal to the sum of pair-wise squared Euclidean distances divided by the total data points.

How to assess best *k*?

The measuring clustering algorithm can be computed in various ways. In this experiment, inertia and silhouette score is used. The inertia is a sum of squared distances of data points to their closest cluster centroid. The lower the average inertia is, the better as it implies a smaller intra-cluster distance. However, it is also important to maximize the inter-cluster distance. The silhouette score computes the mean silhouette coefficient of all data points. The silhouette coefficient first subtracts the mean of the intra-cluster distance from the mean of the nearest-cluster distance. Finally, the subtracted value is divided by the mean of intra-cluster

distance or the mean of nearest-cluster distance whichever has a larger value. The score ranges from -1 to 1. The optimal value is 1 and the worst value is -1. Generally, negative values indicate that a different cluster is more similar to the sample. It can be interpreted that the negative value sample, therefore, is wrongly-assigned to the cluster. The value near 0 means the clusters are overlapping [3]. Finding a number of the cluster that achieves as low an inertia score as possible and as high a silhouette score as possible is the best. One important thing to keep in mind is that the sum of square error will monotonically non-increase as we are using Euclidean distance. The reason is that the K-means algorithm in Euclidean space repeatedly minimizes average squared error. Therefore, a number of k should be carefully selected considering both the inertia score and the silhouette score.

K-means Result

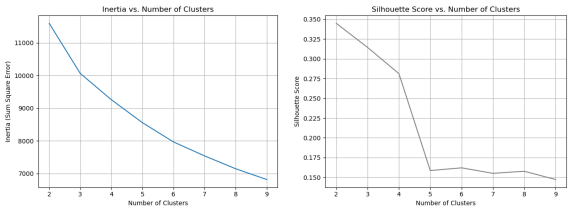

Fig. 1. BWC K-means Scores

Figure 1 presents the result of inertia and silhouette score based on the number of clusters. We can decide the number of k by looking at the knee or the elbow point from the curve. The elbow in the curve entails that adding more clusters does not significantly increase the performance of the model. This approach is widely used in cluster analysis to decide the optimal value for k. From the elbow method perspective, we can infer that WBC returns the best overall scores(inertia and silhouette) when the k is 3. The number of samples in each cluster can be seen in the left-side plot in figure 5. We can assume that majority of samples are clustered in cluster label 0. To better visualize the result, figure 2 illustrates the silhouette score on the left-side plot and the right-side plot shows how clusters are formed in actual 2D space.

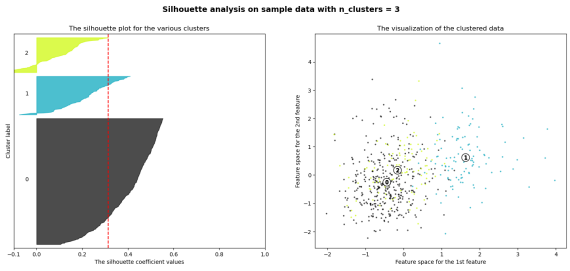

Fig. 2. WBC K-means Visualization

From the right-side visualization of figure 2, we can see that there is a large overlapping space between cluster label 0 and label 2. It would rather make more sense if cluster label 0 contains all of the samples in cluster label 2. The WBC has two labels, which are positive class and negative class. Thus, the cluster does not line up with the true labels. The problem with k being 3 is that the inter-cluster distance is too small that there is almost no distinguishing space between the majority of data points in clusters 0 and 1. It is, of course, important not to force cluster numbers to align with the labels of data. However, it would sometimes make more sense to stop before the elbow point and rely more on the silhouette score when choosing the number of clusters. This approach can be done carefully by both comparing the 2D space visualization, inertia result, and silhouette score to improve the result.

Below figures describes the K-means result on the PID dataset. First, figure 3 shows the inertia and silhouette scores for a different number of clusters. Again, the number of k can be determined by finding the elbow of the curve and the best silhouette score. It seems that the best combination for both scores is when k is 4. Figure 4 visualizes the silhouette score in more detail on the left side. On the right side of figure 4, clusters are presented in 2D space. Unlike WBC, the distribution of data samples themselves is vertically aligned, which makes clustering very hard. This, in part, explains why other machine learning algorithms performed poorly compared to WBC in all of the previous assignments. The problem itself is very difficult to model. Looking at how each sample is scattered, the best way to improve clustering results, or any other algorithms, for this hard problem could be making modifications in the features such as scaling them in a better way or adding useful features that allow the algorithms to model the data more easily. Figure 5 right-side plot shows how many samples are in each cluster. We can see that cluster label 1 has the most data points followed by cluster label 2. Cluster 0 and 3 seems to be just randomly formed area.

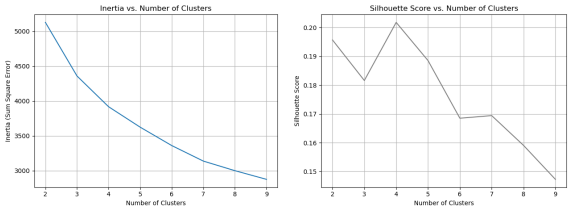

Fig. 3. PID K-means Scores

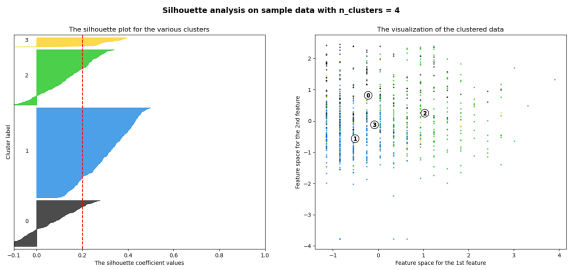

Fig. 4. PID K-means Visualization

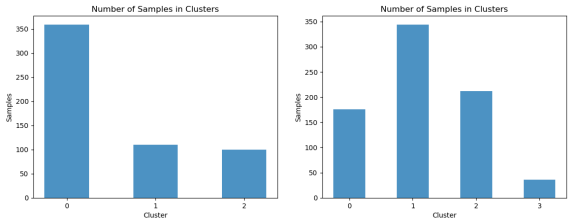

Fig. 5. K-means Sample Size in Clusters

B. *EM*

Given the partial information of the full data, the EM algorithm estimates some set of parameters that describe an underlying probability distribution [4]. EM algorithm iterates the process of the expectation step (E-step) and maximization step (M-step). In the E-step, the algorithm pretends that it knows the parameter of the model and infers the probability that each sample belongs to each component. In M-step, it computes the maximum likelihood hypothesis of the expected probability distribution for the previous step. In this experiment, the Gaussian mixture model is used to perform EM on WBC and PID datasets.

How to assess best *k*?

To choose the best k, three different values are utilized, which are the silhouette score, an Akaike information criterion (AIC), and a Bayesian information criterion (BIC). Both AIC and BIC score a model based on its log-likelihood and complexity. AIC puts more emphasis on model performance, penalizing complex models less. On the other hand, BIC penalizes complex models more and avoids choosing complex models. Thus, both AIC and BIC should be nicely balanced to avoid choosing too complex models or models that are too simple. Both values should be minimized to find the optimal number.

EM Result

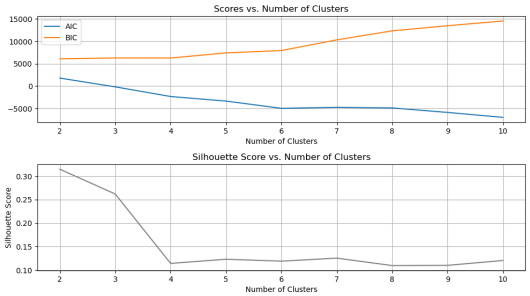

Fig. 6. WBC GMM Scores

Figure 6 presents the three statistics to decide the number of clusters. The best number of clusters is two. As the size of the cluster increases, the silhouette score decreases drastically and BIC values increase. The reason the AIC value decrease is that it favors the model that performs best on the training and validation dataset regardless of the model complexity. Therefore as the model gets complex and the number of clusters increases and improves the accuracy at the cost of clustering is overly complex. This is not desirable as it cannot generalize on the unseen data. When using 2 clusters, cluster label 0 is composed of roughly 210 samples and cluster label 1 is composed of approximately 360 samples (see Figure 8). The result is remarkable because the clusters almost exactly line up with the number of labels in the PID dataset. Moreover, the size of the dataset for each label is also very similar (see Table 1).

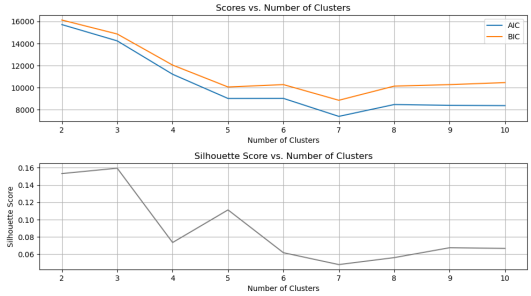

Fig. 7. PID GMM Scores

The GMM performed on the PID dataset is presented in figure 7. The best value for the k, is 3. Increasing the number of clusters lowers the AIC and BIC scores, but decreases the silhouette score as well. Figure 4 right-side plot shows the underlying data distribution in 2D space. More clusters can partition vertically aligned datasets, thus increasing both AIC and BIC. However, the distance between clusters may be too close, which is not a good clustering. The number of data points is shown in the right-side plot of figure 8. It seems like it does not align with the true labels. Similar to the K-means clustering example, the PID has repeatedly been proven by different algorithms that it is a difficult problem set. A feature that can form distinct distribution between two classes can help GMM perform better.

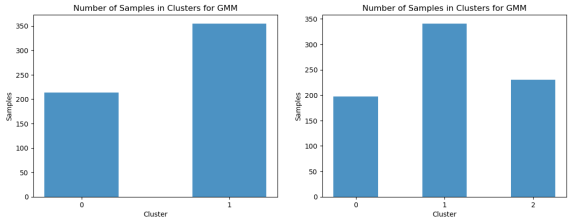

Fig. 8. PID Sample Size in Clusters

C. *K-means vs. EM*

Compared to the K-means clustering method, GMM generated less number of clusters, which is closer to true label distribution. The GMM clustered nearly perfectly on the WBC dataset. Although GMM and K-means both did not produce a matching result with the true label, the GMM clustering result seemed to be more reasonable. EM method may have produced better results compared to K-means with the chosen datasets because it does not force the algorithm

to decide where intermediate boundary variables should go. To be more specific, Gaussians have an infinite extent that even if a point is very far away from the center, it still has some chance of having been generated from that Gaussian. This tells that all points have some non-zero probability of belonging to the other cluster. Approaching problematically to assign the data points to clusters may have contributed to forming more reasonable clusters compared to the distance computing method.

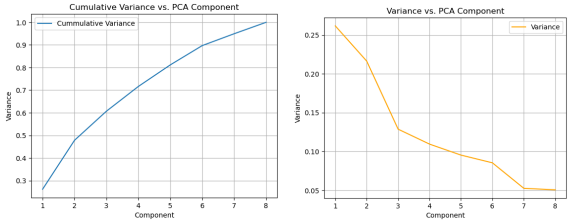

Fig. 10. PID PCA Result

IV. Dimensionality Reduction

In this section, four different dimensionality reduction algorithms are applied to the two datasets. Dimensionality reduction is a technique of pre-processing a set of features to create a new feature set that is smaller or compact while retaining as much relevant and useful information as possible.

A. *PCA*

The principal component analysis takes data and finds a different set of an axis that maximizes the variance and projects the data in those directions. More concretely, PCA is solving the Eigen problem, which repeatedly discards the ones with the least eigenvalues.

How to decide the best number of features?

The explained variance refers to the amount of variance explained by a principal component. In general, the larger the explained variance is, the more important that component is because a larger explained variance means more variance is explained by each component. The explained variance ratio represents the sum of eigenvalues of all eigenvectors as it represents the variance explained using a particular eigenvector [5]. Therefore, the sum of eigenvalues or cumulative sum of variance can be used to decide the principal components that have the most variance or information using the features we have.

PCA best number of *k*

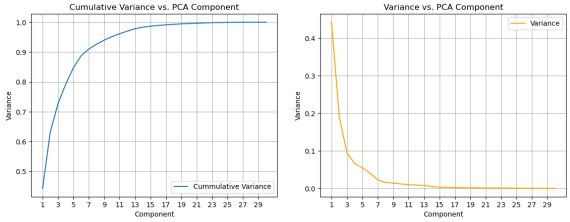

Fig. 9. WBC PCA Result

The above figure presents the cumulative sum of variance(eigenvalues) on the left and individual explained variance on the right. The x-axis is the index of each principal component. We can see from figure 9 that 90% of variances are explained by the 7 components for WBC. After 10 principal components, 95% variance of the dataset is explained. After 10 features, there are only marginal gains. Thus, we can say 10 features are an optimal number of features. The total time to complete the run took 0.022 seconds for the WBC dataset. On the other hand, the best number of principal components is 7 or 8 (see figure 10). This means that all features were required to explain 95% variance of the dataset. Considering that there are only 8 features in the PID dataset, we can assume that every feature plays a crucial role in assigning the label. This alone also tells that adding more useful features can help algorithms to perform better in unsupervised learning tasks.

PCA Reconstruction Error

Lastly, the reconstruction error, which indicates the distance between the projected data point onto a lower-dimensional subspace and the original point. This is the error caused by projecting data to a lower dimension. Thus, minimizing the reconstruction error is desirable. The reconstruction error for WBC is 1.2e-30 and 2.6e-31 for PID. The reconstruction error for both datasets is low when applying PCA. Although PID had a lower reconstruction error, this largely has to do with the algorithm using all the features given whereas less than one-third of features were used for WBC. Hence, we can assume that WBC is a relatively easier problem compared to PID and also has key features that determine the classes, unlike PID dataset.

PCA Run Time

The total time to complete the run was 0.003 seconds for the WBC dataset and 0.001 for the PID. Although the PID dataset is larger, we can see that number of features contributed to a longer run time for PCA.

B. *ICA*

Independent component analysis, in contrast to PCA, tries to maximize independence. To be specific, the ICA attempts to find a linear representation of non-Gaussian data and project it into a new feature space such that each of the individual new features is statistically independent.

How to decide best number of features?

The two underlying assumptions of ICA are: the sources are mutually independent of one another and the independent component must have non-Gaussian distributions. It is proven that Gaussian variables can estimate the ICA model only up to an orthogonal transformation [6]. This entails that maximizing non-Gaussianity is independent and, therefore, gives us independent components. Thus, we can use kurtosis as an optimization criterion since kurtosis is the classical measure of non-Gaussianity. The higher the kurtosis value is, the more super-Gaussian it is. The super-Gaussian typically is a spiky

probability density function with heavy tails. The kurtosis is usually nonzero for non-Gaussian random variables.

ICA best number of components

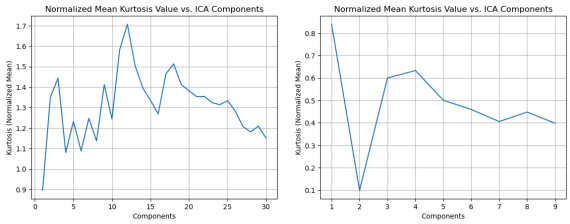

Fig. 11. ICA Results for Two Datasets

Figure 11 displays the best number of independent components for both datasets. The left-side plot from figure 11 is the result of the WBC dataset. It is estimated that a total of 12 independent components are the best for WBC. For the PID dataset, just one independent component records the highest kurtosis. We can infer that when the components are at 12 for WBC, the distribution is fairly kurtotic because the normalized mean of kurtosis is slightly above 1.7. The PID dataset reached almost 0.85 for the kurtosis mean, which also infers that the distribution is less kurtotic compared to PID. Nevertheless, it still is a non-Gaussian distribution as the kurtosis is a non-zero value.

ICA Reconstruction Error

The reconstruction error for WBC is 0.029 and 0.738 for PID. Reconstruction error is high for PID, indicating the difficulty of finding the independent component while minimizing the information loss. We can assume that although the ICA generated the most non-Gaussian distribution, the PID dataset was too complex for ICA to perform well unlike the WBC dataset.

ICA Run Time

The time it took to compute ICA for WBC was 1.124 seconds and 0.036 seconds for PID. The PCA and ICA algorithms take features and project data into a lower-dimensional subspace. Thus, the number of features determines the compute time. Similar to the PCA result above, the WBC dataset took more than 30 times to complete the run compared to the time it took for the PID dataset.

C. *RP*

Randomized projection is also a technique to reduce the dimensionality in Euclidean space. In contrast to PCA, RP selects any vector or direction and performs projection instead of calculating a direction that maximizes the variance. As a result, RP is computationally efficient, fast, and performs well in high-dimensional space. In this experiment, Gaussian random projection, which constructs the projection matrix via a Gaussian distribution, is applied to the two datasets.

RP best number of features For the same reasons explained in ICA, kurtosis will be used to determine the best number of components. Figure 12 illustrates the result for WBC on the left side of the graph and the PID result on the right side of the graph. For both WBC and PID datasets, the highest normalized mean of kurtosis is when the number of components is 2. However, it is important to rerun the algorithm multiple times for RP as it randomly selects any direction to perform the projection.

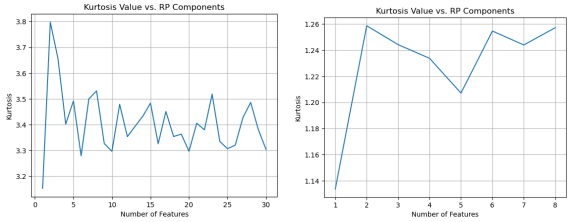

Fig. 12. RP Results for Two Datasets

Re-running RP Result

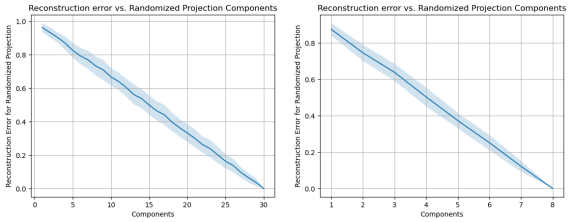

Fig. 13. RP Repeated Results for Two Datasets

After re-running 100 times for the same RP algorithm, the reconstruction error was approximately 0.97 for WBC and 0.88 for PID. Figure 13 presents the summarized result of 100 iterations. The reconstruction error is the worst among other algorithms. The worst performance can be explained by the random nature of RP. As RP takes any direction instead of maximizing variance or independence, it may continually make non-optimal decisions. The lightly shaded fill space in figure 13 shows the number of variations for each run. We can see that as the number of components gets closer to all the available features, there are fewer variations. However, we could also see that the variation could be as large as 7% to 9%.

RP Run Time

The total time it took to re-run the RP 100 times was around 2 seconds for the WBC dataset and around 0.29 seconds to apply RP on the PID dataset. This means that for each run, RP would take roughly 0.002 seconds for WBC and 0.0029 seconds for PID. This is much faster than the ICA running time.

D. *UFS*

The univariate feature selection used univariate statistical tests such as F-test to measure the importance of each feature. UFS estimates the degree of linear dependency between the feature and the label. We can measure the degree of this dependency via the univariate scores that UFS returns for each

feature.

UFS best number of features

Figure 14 visualizes the univariate score for each feature. These univariate scores or p-values tell the significance of the feature in determining the output variable. As expected, many features in WBC play an important role compared to the PID dataset, which is plotted on the right side of the figure. For the WBC dataset, we can say 7 features have the most strong relationship with the output variable and four for PID.

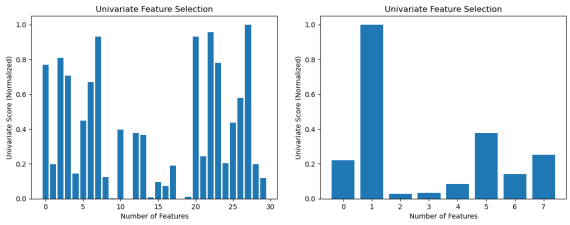

Fig. 14. UFS Results for Two Datasets

UFS Limitation

The downside of the UFS is that it does not consider the interaction between features. In other words, the UFS only considers individual features separately to decide how important the feature is. In the real world, even a simple problem is the result of multiple attributes. Hence, the result of UFS may not be completely reliable.

UFS Run Time

The total time it took for UFS on the WBC dataset was 0.204 seconds and 0.153 seconds on the PID dataset. Because it is assessing each feature separately one by one, it may linearly increase as the number of features increases.

V. Clustering on Reduced Dimension

In this section, clustering experiments are reproduced on the datasets projected onto the new spaces created by above dimensionality reduction algorithms.

A. WBC Dataset Results

K-means

In the previous clustering section, the best number of k was three with the WBC dataset. Therefore, K-means with three clusters are applied to the reduced-dimension dataset. Table 2 shows the summarized result. None in table 2 refers to the original dataset without any dimensionality reduction. Inertia, or the sum of squared distance between data points in the intra-cluster is lowest with ICA applied dataset. However, the silhouette score is the lowest. It shows that ICA-based clusters formed clusters that have a small distance between different clusters. This is not an optimal form of cluster results because it entails that there may be lots of overlapping spaces between different clusters. In contrast, UFS results in the highest silhouette score and lowest inertia score. In addition, it took the least amount of time compared to other datasets. Considering both inertia score, silhouette score, and run time, UFS is the best performing algorithm followed by PCA.

TABLE II
K-means on Reduced Dimension WBC Result Table

Algorithm	Inertia	Silhouette	Time(s)
None	10061	0.314	0.07
PCA	8534	0.338	0.05
ICA	10	0.132	0.07
RP	12785	0.386	0.06
UFS	906	0.459	0.04

Figure 15 visualizes the inertia and silhouette score for each number of clusters. We can find out from this figure the best number of k for each dataset with a different dimensionality reduction algorithm. When the k is 3, it seems to be the best number for the same reason explained in all of the previous sections. The chosen number of k is the same number of clusters as before. The result indicates that the reduced-dimension dataset with all algorithms can achieve the same clustering results with much fewer features.

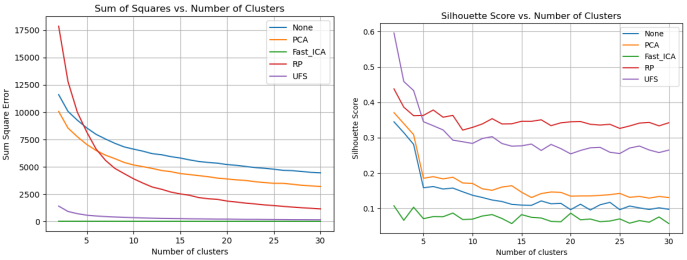

Fig. 15. K-means Results after Dimensionality Reduction on WBC

EM

In the previous clustering section, the best number of k was 2 with the WBC dataset when applying the EM algorithm. The EM with two clusters is applied to the reduced-dimension dataset to compare the results of different algorithms. Table 3 is the summarized result. Overall, the ICA had the lowest AIC

TABLE III
EM on Reduced Dimension WBC Result Table

Algorithm	AIC	BIC	Silhouette	Time(s)
None	1789	6094	0.314	0.02
PCA	13956	14265	0.313	0.01
ICA	-25335	-24549	0.216	0.01
RP	6503	6551	0.449	0.01
UFS	-3424	-3115	0.520	0.01

and BIC scores followed by UFS. The UFS, however, resulted in a silhouette score that is more than double the score for ICA. The time it took for both algorithms is the same. Therefore, both models can be optimal. The worst-performing model was PCA compared to the other algorithms. This could be the result of removing crucial information by performing dimensionality reduction. Figure 16 illustrates three different scores to explain the result as the number of clusters increases. Comparing all

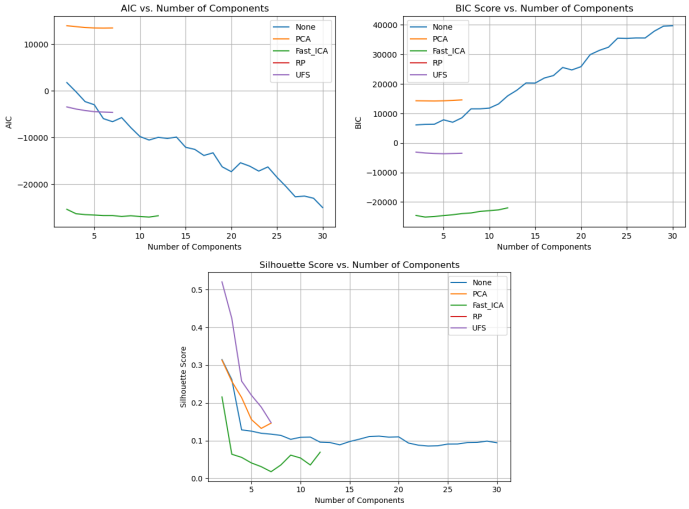

Fig. 16. EM Results after Dimensionality Reduction on WBC

three plots in figure 16, the best number of the cluster is 2 for the same justifications used in previous sections. The result, a gain, aligns with the EM result on the original dataset. We can believe that the reduced dimension dataset still contains useful information to derive the same conclusion.

B. PID Dataset Results

K-means

The same experiment was conducted on the PID dataset. The best number of clusters was 4 in the prior experiment for K-means clustering. Four different dimensionality algorithms were applied to the PID dataset. Then, K-means with three clusters are applied in those datasets. Table 4 displays the result of the experiment. The ICA algorithm produced the best inertia and silhouette score followed by RP and UFS. Although ICA did not perform well in the WBC dataset in terms of silhouette score, it is doing well in the PID dataset. One reason to explain this is that the underlying data distribution is more independent compared to the WBC dataset, resulting in better modelling results. A similar explanation applies to the UFS result. Since UFS takes into account only the strength of independent features on the output variable, a dataset with less correlated features can produce a better result.

TABLE IV
K-MEANS ON REDUCED DIMENSION PID RESULT TABLE

Algorithm	Inertia	Silhouette	Time(s)
None	3918	0.202	0.06
PCA	3608	0.213	0.06
ICA	0	0.532	0.06
RP	1847	0.329	0.05
UFS	1507	0.240	0.05

Figure 17 shows the result of running K-means on the dimension-reduced PID dataset. We can see that the elbow point is at three for all algorithms for the inertia score, which

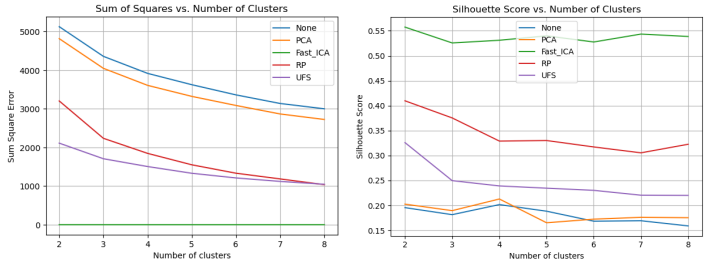

Fig. 17. K-means Results after Dimensionality Reduction on PID

is plotted on the left side of figure 17. The silhouette score is reduced at three clusters. However, considering the high inertia score with two clusters, three clusters generate the best-balanced result. The initial value was four clusters without dimensionality reduction. After reducing the dimension, three clusters are considered as an optimal number. This is more close to the true classes we have for PID, which is two. This may be the effect of removing unnecessary dimensions. Therefore, with less number of features, the EM model can create an improved result.

EM

Table 5 summarizes the result of conducting the EM algorithm after dimensionality reduction on the PID dataset. In the previous experiment, three clusters were the best number of clusters. Applying the EM algorithm with three components on newly created data, the ICA algorithm repeatedly produces the most generalized model. It scores noticeably low AIC and BIC, which is desirable. Also, it has the highest silhouette score followed by RP and UFS. The intuition behind these well-performing models is explained in the above K-means part. This experiment conveys the importance of knowing the characteristics of the dataset. For example, knowing whether the dataset has a strong independency between features, we can choose an algorithm accordingly.

TABLE V
EM ON REDUCED DIMENSION PID RESULT TABLE

Algorithm	AIC	BIC	Silhouette	Time(s)
None	14241	14863	0.160	0.02
PCA	13939	14436	0.138	0.01
ICA	-2918	-2881	0.504	0.02
RP	6036	6115	0.350	0.01
UFS	7485	7689	0.134	0.04

Figure 18 depicts the best number of components based on AIC, BIC, and the silhouette score. All plots omit the score for just one component. The elbow of both the AIC and BIC graph is at four for the number of components, the silhouette score drops significantly at four components. The silhouette score is higher when the number of components is two. Therefore, the optimal number of the component would be 2 or 3, which is closer to the ground-truth label.

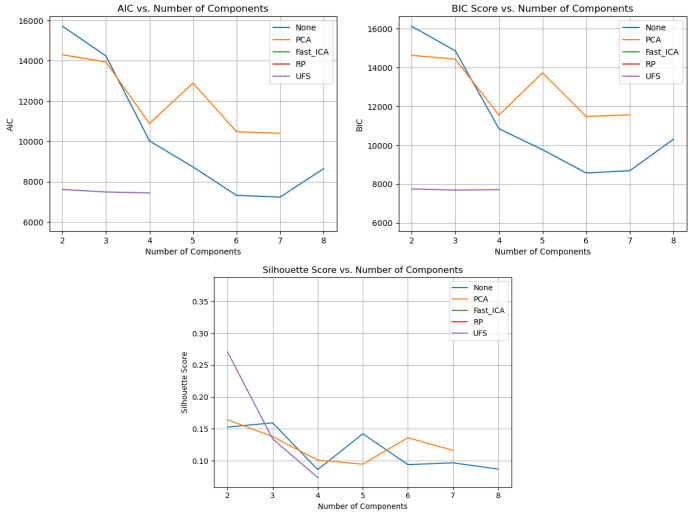

Fig. 18. EM Results after Dimensionality Reduction on PID

VI. Neural Network with Dimensionality Reduction

The four different dimensionality reduction algorithms will be applied to create newly projected data. Then, a neural network will be applied to those datasets. For comparison, figure 19 provides the result of running a neural network on the original dataset without any dimensionality reduction. The best hyperparameter configuration is found by 5-fold cross-validation. In addition, increasing layer size too much when the dataset size is small can increase the risk of overfitting. Thus, the search space for the number of layers is constrained to a maximum of two since we have a small dataset.

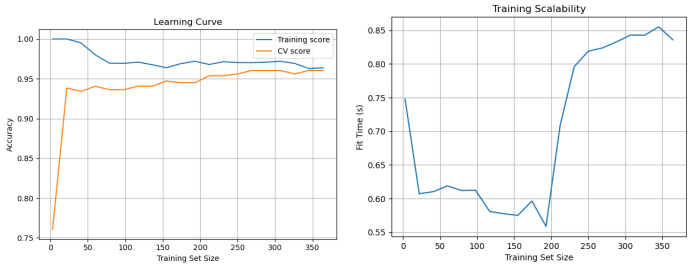

Fig. 19. Neural Network with the Original WBC Dataset

Below table 6 lists the final model architecture with the chosen hyperparameter set. Also, table 6 contains the total fit time and the accuracy score. The layer size indicates that the best number of layers chosen is two with a hidden node size of 40 for the first layer and 10 for the second layer for the PCA dataset. The constant learning rate uses 0.01 as the default value. For each dimensionality reduction algorithm, the chosen best number of components are same as the previous result. Detailed analysis is continued in each subsection.

TABLE VI
Neural Network with Dimensionality Reduction Result

Algorithm	Layer Size	Learning Rate	Optimizer	Time(s)	Accuracy
None	(20,)	constant	adam	37.57	0.980
PCA	(40, 10)	constant	sgd	33.78	0.976
ICA	(20, 10)	constant	sgd	38.08	0.976
RP	(30,)	constant	lbfgs	23.25	0.662
UFS	(20,)	constant	lbfgs	24.29	0.960

A. PCA

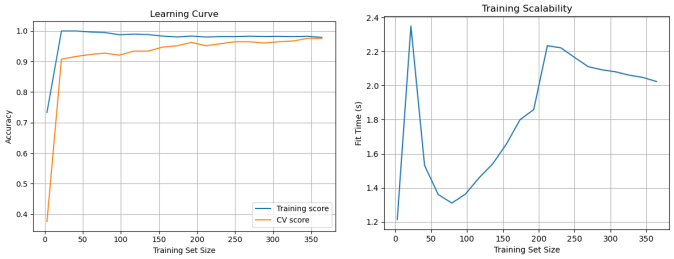

Fig. 20. Neural Network with the PCA

When running a neural network on a PCA dataset, the accuracy is the same when rounded. Although it needs one more layer to produce a similar result, the time it took for PCA based dataset is approximately four seconds faster. Figure 20 shows the learning curve and training scalability. The variance seems to be decreasing as the number of training example increase, meaning there is no sign of overfitting. Due to the reduced number of dimensions from 30 features to 7 features, the training scalability is quite different from the original dataset. To summarize, the neural network with the PCA dataset produced a very generalizable model with almost no difference in accuracy compared to the original model.

B. ICA

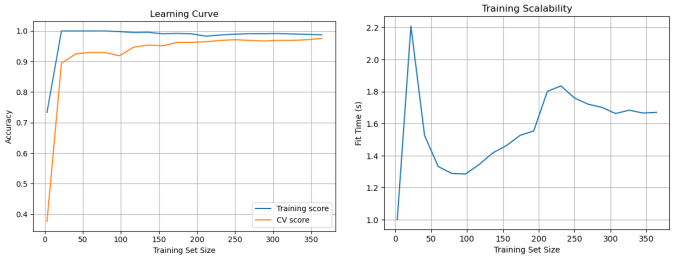

Fig. 21. Neural Network with the ICA

The neural network with the ICA dataset took roughly 0.5 seconds more compared to the original result but, again, produced almost the same accuracy score as the original model. Similar to PCA, ICA variance is very small, showing no sign of overfitting in figure 21. During ICA, a total of 30 features were reduced to 12 dimensions. Therefore the

training scalability is dissimilar to the original graph but shows a similar pattern as the PCA dataset. The ICA-based dataset also generated a strong model that generalizes well in a much lower dimension compared to the original 30-feature dataset.

C. RP

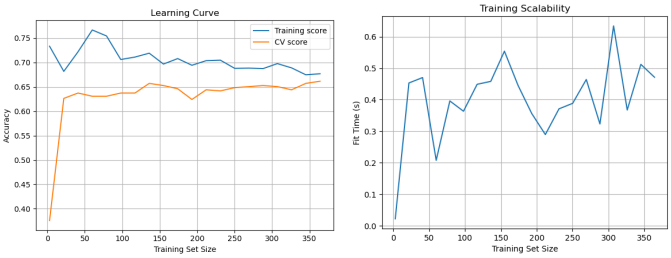

Fig. 22. Neural Network with the RP

RP-based dataset resulted in the lowest accuracy score compared to all four different methods. The accuracy score dropped to 0.662, although it took significantly less time compared to all the previous methods. As explained in-depth in the previous sections, the RP algorithm randomly picks the direction instead of the direction that maximizes the variance or independence. As a result, it is vulnerable to making bad decisions. However, RP is a very fast algorithm that generates ok and sometimes excellent results as could be seen in the previous sections. RP could be very useful in settings with limited time and resources. The learning curve from figure 22 illustrates that there is a clear sign of overfitting. Even if the training set size increases, the gap between training and cross-validation score does not narrow down. The model is doing better with the training dataset, but constantly doing worse with the validation dataset, which means it is not generalizing well to the unseen dataset. The Training scalability doesn't show many variations because the number of dimensions was only 2 for RP based dataset.

D. UFS

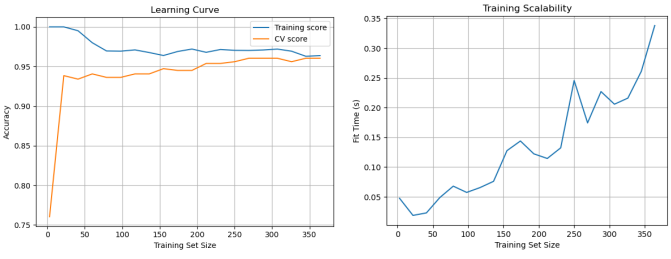

Fig. 23. Neural Network with the UFS

The neural network with UFS applied dataset took less than 10 seconds compared to the original model. It is also the fastest model among all the others. Nevertheless, there is a slight decrease in accuracy. Yet, considering its fast speed, the loss in accuracy is trivial. The final accuracy score was 0.96, which is 0.02 lower than the original dataset. For UFS applied dataset was reduced to 7 dimensions. Compared to PCA and ICA, the UFS-based dataset model shows more variance (see figure 23). The variance decreases as the number of training examples increases. The overfitting is not serious here, but clearly suffers more compared to PCA and ICA and hence resulted in lower accuracy. It would be a better option compared to PCA, ICA, or the original model when you want to run the model fast at the cost of a little decrease in accuracy.

VII. Neural Network with Clustering as Dimensionality Reduction

The newly created dataset with clustering is fed into the neural network as the training dataset. Original features are now reduced to the number of clusters produced from the clustering algorithm, which is three for K-means and two for the EM algorithm. Table 7 presents the result of neural networks based on the clustering algorithm. The detailed explanation is continued below.

TABLE VII
Neural Network with Clustering Result

Algorithm	Layer Size	Time (s)	Accuracy
None	1	37.57	0.980
K-means	1	37.493	0.976
K-means	7	124.48	0.99
EM	1	76.569	0.624
EM	5	139.04	0.99

A. K-means

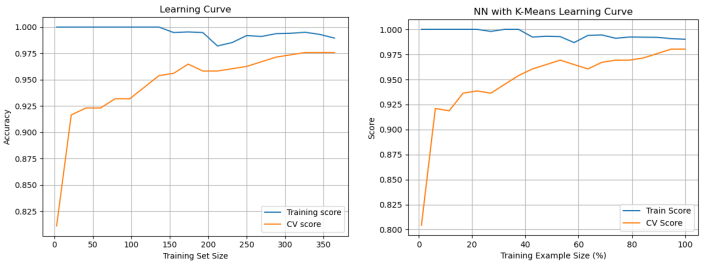

Fig. 24. Neural Network with the K-means

Figure 24 is the learning curve for the neural network run on K-means applied dataset. The left-side plot is the smaller layer size, which is 1. The right-side plot is the learning curve for 7 layers neural network. The accuracy for a one-layer neural network has almost the same accuracy score with the roughly same fit time. The learning curve also suggests that a one-layer neural network is suffering from overfitting. The model is constantly doing better on the training date versus the validation dataset. However, the discrepancy in performance between the training and validation dataset is not too severe. With just a one-layer neural network, we can achieve almost the same result with a much lower dimensional

dataset compared to the original higher dimensional dataset. The right-side plot is the learning curve for a larger network. The best number of layers found by hyperparameter tuning was seven layers, which outputs better results than both the original model and the one-layer neural network. The learning curve also shows a similar result to the simpler model. It has a high variance issue but is reasonable enough to still adopt in replace of the original model. However, the fit time is more than three times slower for the bigger neural network. The benefits of dimensionality reduction are shortening time and lowering variance. The seven-layer neural network does not have these advantages.

B. EM

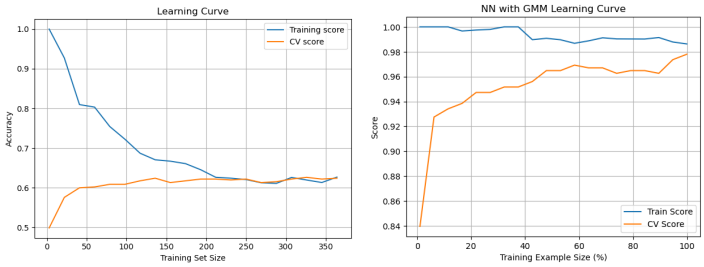

Fig. 25. Neural Network with the EM

The same experiment is now tested on the EM algorithm. Figure 25 shows a neural network with one layer on the left side and a neural network with 5 layers on the right side of the figure. Both models used the EM algorithm as a method of reducing the dimensions of the original dataset. The best number of components found in the previous experiment was used, which was two. Examining the results in table 6, the neural network with one layer seems to output a poor accuracy score. This is a noticeably big difference compared to all the other models. The learning curve, on the other hand, shows that the model is underfitting. The learning curve displays low training and validation accuracy scores even if the training set size increases. In this case, the one-layer neural network has a low capacity to handle the complexity of the given dataset. We can make a more complex model to be able to express rich data representation. The neural network with five layers, which was discovered by the hyperparameter search, marks good results. The final best model accuracy was 0.99 for the 5-layer neural network. The model, however, seems to suffer from an overfitting. As the training example ratio increases, the variance slowly decreases. Again, the difference ratio would be roughly 5%, which is acceptable. The downside of increasing the layer size is that it drastically slows down the model fit time. The overall analysis demonstrates the accuracy and time trade-off.

VIII. Conclusion

In this analysis report, five different experiments were conducted. Specifically, clustering and dimensionality reduction algorithms were explored in-depth. The comprehensive takeaway from these experiments is that dimensionality reduction and clustering can sometimes find useful information from the data that are often not easily visible.

References

[1] UCI Machine Learning Repository: Breast Cancer wisconsin (diagnostic) data set. (n.d.). Retrieved August 28, 2022, from https://archive.ics.uci.edu/ml/datasets/

[2] Pima Indians Diabetes Database. (2016, October 6). Kaggle. Retrieved August 28, 2022, from https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

[3] Sklearn.metrics.silhouettescore. scikit. (n.d.). Retrieved October 29, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouettescore.html

[4] Machine Learning. (1997). McGraw-Hill Science/Engineering/Math.

[5] Kumar, A. (2022, August 11). PCA Explained Variance Concepts with Python Example. Data Analytics Data, Data Science, Machine Learning, AI. Retrieved October 30, 2022, from https://vitalflux.com/pca-explained-variance-concept-python-example/

[6] Hyvärinen A, Oja E. Independent component analysis: algorithms and applications. Neural Networks : the Official Journal of the International Neural Network Society. 2000 May-Jun;13(4-5):411-430. DOI: 10.1016/s0893-6080(00)00026-5. PMID: 10946390.