Name: Sol Won

Machine Learning - Unsupervised Learning

Description of the dataset:

Two datasets are used to analyze different machine learning algorithms. (Breast cancer Wisconsin data and Diabetic Retinopathy Debrecen dataset) Two datasets are the same datasets used for Assignment 1 and 2.

Breast Cancer Wisconsin (Diagnostic) Dataset: There are total of 30 attributes of cell nucleus and they are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. Classification problem is to predict whether the tissue is benign or malignant by looking at those 30 attributes. The reason of using this dataset is as follows. First, there are lot of attributes in the dataset. For instance, there are three types of radius which are radius SE, radius mean and radius worst. This made the dataset attractive in a sense that there might be tons of different results or changes that can happen. Moreover, the dataset is well distributed. There are 357 benign cases and 212 malignant cases. The dataset is not perfectly balanced (which is 50% for each class) but good enough to be applied and analyzed.

Diabetic Retinopathy Debrecen Dataset: This dataset consists of attributes from Messidor image to predict whether an image has diabetic retinopathy sign or not. There are total of 18 attributes from the image. Classification problem is to predict whether the image has sign of diabetic retinopathy or not. This dataset is used since it is well balanced. There are 540 positive cases and 611 negative cases. Also, I personally found this dataset interesting since I always wonder about how well I can train the information based on the image.

Clustering algorithms:

K-means clustering and Expectation maximization from Gaussian Mixture Model (GMM) are used for this analysis. Scikit learn library is used to represent those to clustering algorithms. Clustering itself is defined as the most common technique in data analysis where it separates the data and group the similar ones into subgroups. K-means and Expectation maximization is one of them.

K-means algorithm is defined as grouping of given dataset into k number of clusters. Each data is separated to the cluster with the nearest mean. Algorithm tries to minimize the variance of distance differences. Moreover, algorithm gives labels to the unlabeled input datasets. To select and choose the right number of clusters, I ran K-mean algorithm with different number of clusters (range 2-11).

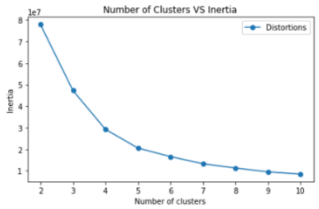

Figure 1: # of Clusters VS Inertia

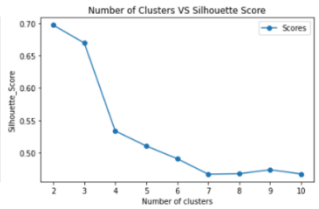

Figure 2: # of Clusters VS Silhouette

Figure 1 and 2 are used to determine which k clusters to choose for the first dataset. Combination of Elbow method and Silhouette score is used to determine the k value. For figure 1, as number of clusters increase, the cost (inertia) continuously decrease. This is because the distance between centroid and

data points decrease. Graph has an elbow shape line where the cost decreases a lot at the beginning and slows down as it goes. In elbow method, we choose the cluster where the rate of decrease slows down. Therefore, using 5 clusters is optimal. Figure 2 shows the Silhouette score at each cluster. Silhouette score shows how similar the data points in the cluster are compare to the other clusters. It is better to have cohesion between the data points within the cluster and separation with other clusters. -1 to 1 is the range of the score and it is better to have a high score. For dataset 1, the average silhouette score is 0.5306. (Max: cluster 2 with 0.6973, Min: cluster 10 with 0.4669) By looking at the graph, clusters 2, 3, 4, 5 is above the average. Cluster 5 is chosen as optimal after applying both methodologies.

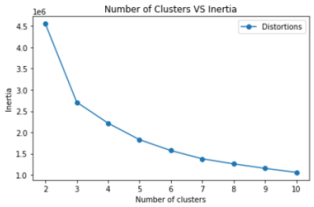

Figure 3: # of Clusters VS Inertia

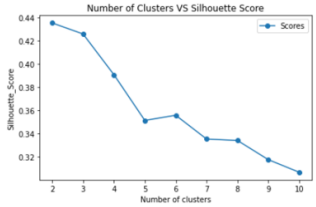

Figure 4: # of Clusters VS Silhouette

Figure 3 and 4 are used to determine which k clusters to choose for the second dataset. Figure 3 shows that the rate of decrease slows down around cluster 4 and 5. For silhouette score, the average is 0.3613. (Max: cluster 2 with 0.4352, Min: cluster 10 with 0.3065) Clusters above the average are 2, 3 and 4. Cluster 4 shows optimal performances after applying both methodologies.

Expectation Maximization (EM) is defined as finding the optimal parameter by using two steps which are E-step and M-step. E-step calculate the closer likelihood from the initial parameter likelihood. M-step maximizes the calculated likelihood and produce new parameter value. Those two steps iterate continuously. Gaussian Mixture Model (GMM) is one of the models that apply EM algorithm. Scikit learn library is used to implement GMM.

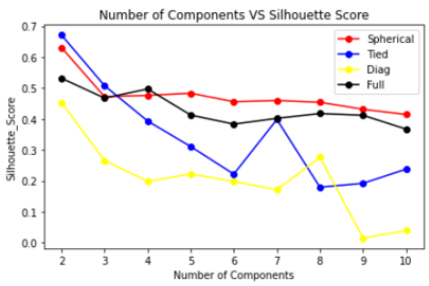

Figure 5: # of Components VS Silhouette

Figure 5 shows the silhouette score of 4 different covariance types of dataset 1. Spherical, tied, diag, and full is used and as shown on the graph, spherical covariance type has the highest average silhouette score. For silhouette score, the average is 0.4751. (Max: cluster 2 with 0.6305, Min: cluster 10 with 0.4143) Clusters above the average are 2, 4 and 5. Optimal cluster to use by looking at the silhouette score is 5. Another reason why spherical analysis was chosen is because K-mean only detects spherical clusters.

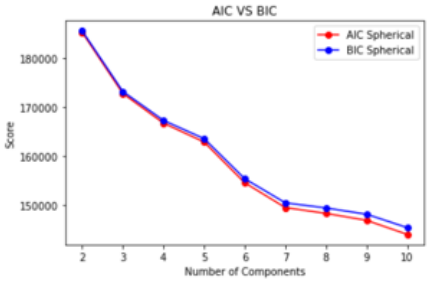

Figure 6: # of Components VS Score

Aikaki Information Criterion (AIC) and Bayesian Information Criterion (BIC) can also be used to specify the number of components. It is better to use AIC and BIC together, but the difference is that BIC is stricter and penalizes heavily compare to AIC. The number (score) on the y-axis is not that important. The lower the better. Figure 6 was interesting since the score drops as number of components increased. Component 10 had the lowest score among all the other clusters. Figure 5 shows that there is not much of a difference of silhouette score between cluster 5 to 10. The decision of which number of components to use depends on what to put more weight on.

Since there is a huge gap of score between component 5 and 10 in AIC/BIC graph, cluster 10 is used for the future analysis.

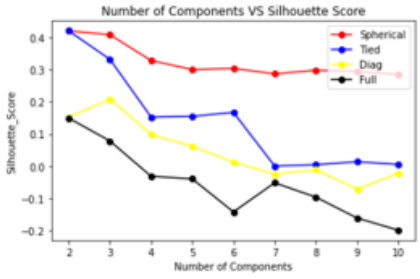

Figure 7: # of Components VS Silhouette

Figure 7 shows the silhouette score of 4 different covariance types of dataset 2. Spherical covariance type has the highest average silhouette score. For silhouette score, the average is 0.0.3251. (Max: cluster 2 with 0.4204, Min: cluster 10 with 0.2852) Clusters above the average are 2, 3 and 4. Optimal cluster to use by looking at the silhouette score is 4. Other covariance types have low silhouette score compare to spherical covariance.

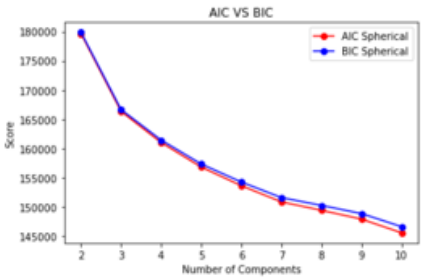

Figure 8: # of Components VS Score

AIC and BIC is used to find the best optimal number of components. Similar with figure 6 (with dataset 1), componennt 10 had the lowest AIC and BIC score. Figure 7 shows that clusters from 4 to 10 have similar silhouette scores. Compoent 10 seems reasonable to make it an optimal component.

Now, dimensionality reduction algorithms will be applied to each dataset and be clustered by K-mean and GMM.

Clustering with Dimensionality reduction algorithms (PCA):

Data in real life that we analyze has lots of features. If we are to solve a problem with machine learning algorithms, there is a high possibility that the performance will not be optimized. As the number of features in the dataset increases, dimension also increases. And as a result, density will be sparse. Principal component analysis (PCA) is one of the popular dimensionality reduction algorithms. PCA focus on finding principal components while maximizing the variance.

K-mean Clustering and GMM with PCA (Dataset 1):

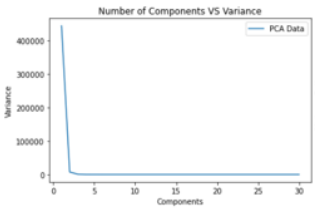

Figure 9: # of Components VS Variance

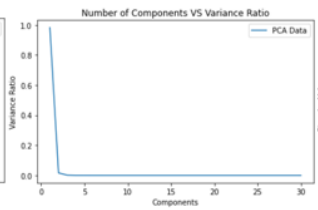

Figure 10: Components VS Variance Ratio

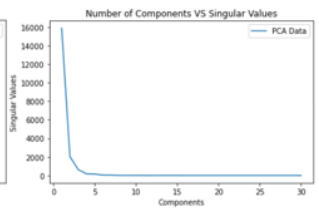

Figure 11: Components VS Singular Values

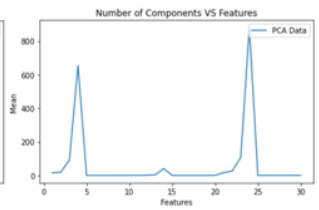

Figure 12: Components VS Means

Figure 9 to 10 is used to estimate how many components are needed to obtain and reach 100% of the variance in dataset 1. Figure 9 is a variance graph where there is a drop between component 2 and 3. Little bit more decrease after that and it reaches 0 around component 7. Figure 10 shows the ratio of the variance. The curve looks pretty much the same with figure 9. By looking at the graph, 100% of the variance is achieved at component 6 or 7 (cumulative variance ratio). Figure 10 shows the singular values corresponding to the components (eigenvalue). It has elbow shape as well but unlike other graphs, decreases until it reaches component around 5. Therefore, component of 5 will be used for later analysis. Lastly, the mean is calculated. This is the feature

empirical mean which is estimated from the training set. PCA assumes all the data to have a zero mean. It is because when we are calculating for PC, it is being calculated bias on mean.

K-mean clustering and GMM is done after applying PCA algorithm. PCA Component 5 is used.

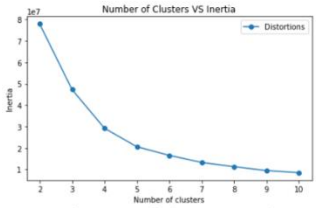

Figure 13: # of Clusters VS Inertia

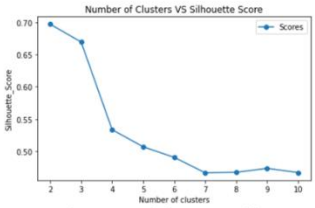

Figure 14: # of Clusters VS Silhouette

Figure 13 and 14 are used to determine which k clusters to choose for dataset 1. Figure 13 shows that the rate of decrease slows down around cluster 4 to 5. For silhouette score, the average is 0.5302. (Max: cluster 2 with 0.6973, Min: cluster 10 with 0.4669) Clusters above the average are 2, 3 and 4. This graph looks very similar with the original K-means clustering graph with the original data. Without PCA, the average was 0.5306 and 0.5302 with PCA.Which in this case means PCA was successful reducing the dimension and at the same time, produce satisfying silhouette score.

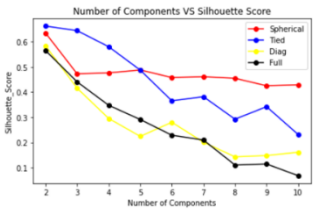

Figure 15: # of Components VS Silhouette

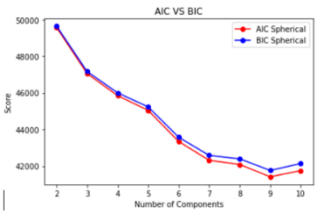

Figure 16: # of Components VS Score

For GMM, spherical covariance type still had the highest average silhouette score compare to other types. For silhouette score (figure 15), the average is 0.4773. (Max: cluster 2 with 0.6340, Min: cluster 10 with 0.4286) Clusters above the average are 2 and 5. Optimal cluster to use by looking at the silhouette score is 5. Interesting fact is that silhouette score actually increased compare to the original silhouette score which was 0.4751 average. Figure 16 shows the AIC and BIC score. Component 9 had the lowest score and the score went up after that (component 10). Optimal component in this case is 10 since silhouette score from component 3 to 10 is very similar but there is a huge difference in terms of AIC and BIC score.

K-mean Clustering and GMM with PCA (Dataset 2):

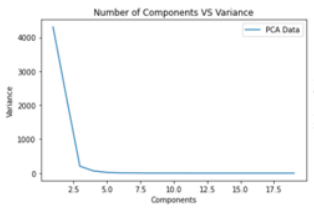

Figure 17: # of Components VS Variance

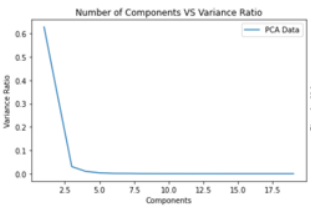

Figure 18: Components VS Variance Ratio

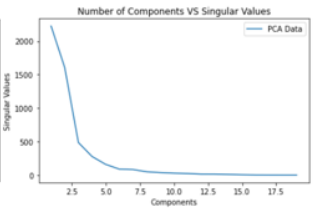

Figure 19: Components VS Singular Values

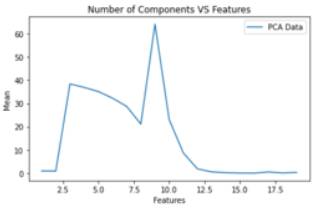

Figure 20: Components VS Means

Figure 17 to 18 is the graph of variance for dataset 2. Figure 17 is a variance graph where there is a significant decrease between component 2 and 3. Little bit more decrease after that (until component 5) and it reaches 0 around component 7. Figure 18 shows that 100% of variance is achieved at component 7. Figure 19 has elbow shape as well, but the eigenvalue decreases until it reaches component around 10. Therefore, component range of 7 to10 will be used for later analysis.

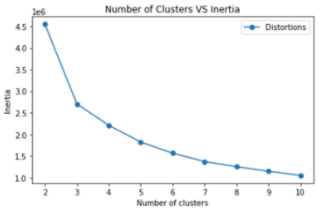

Figure 21: # of Clusters VS Inertia

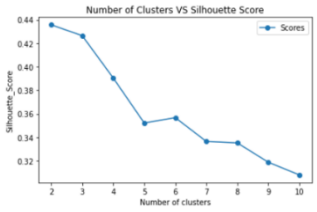

Figure 22: # of Clusters VS Silhouette

K-mean clustering and GMM is done after applying PCA algorithm for dataset 2. Components of 7 is used. Figure 21 and 22 are used to determine which k clusters to choose for dataset 2. Figure 21 shows that the rate of decrease slows down around cluster 3. For silhouette score, the average is 0.3622. (Max: cluster 2 with 0.4356, Min: cluster 10 with 0.3080) Silhouette score after applying K-mean increased from 0.3613 to 0.3622. Cluster 4 is the optimal cluster by looking at two graphs.

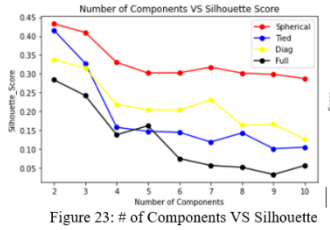

Figure 23: # of Components VS Silhouette

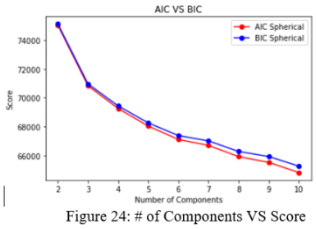

Figure 24: # of Components VS Score

For GMM, spherical covariance type had the highest average silhouette score compare to other types. For silhouette score (figure 15), the average is 0.3312. (Max: cluster 2 with 0.4325, Min: cluster 10 with 0.2873) Optimal cluster to use by looking at the silhouette score is 7. Interesting fact is that silhouette score actually increased compare to the original silhouette score which was 0.3251 average. By looking at Figure 24, Component 10 had the lowest score and the score went up for component 10. Optimal component in this case is 10 since silhouette score from component 3 to 10 is very similar but there is a huge difference in terms of AIC and BIC score.

Clustering with Dimensionality reduction algorithms (ICA):

Independent Component Analysis (ICA) is defined as statistical procedure to change raw data into sets of values of independent variables. This can be helpful to identify similar patterns within the dataset.

K-mean Clustering and GMM with ICA (Dataset 1):

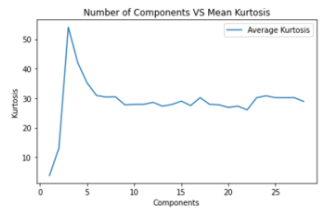

Figure 25: # of Clusters VS Kurtosis

Figure 25 is the Kurtosis graph that is used to figure out the number of components to use in ICA for dataset 1. After calculating the mean of Kurtosis for each component, component 4 has the max kurtosis. If there is a high kurtosis, we can assume that there are lot of outliers which can be viewed as a noise. After component 4, mean kurtosis drops and the line becomes steady. Optimal component of 5 is used for further analysis.

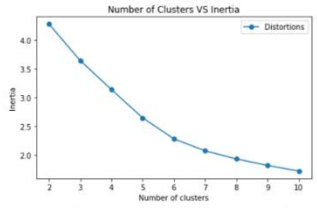

Figure 27: # of Clusters VS Kurtosis

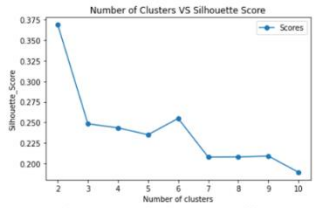

Figure 28: # of Clusters VS Silhouette

K-mean clustering and GMM is done after applying ICA algorithm for dataset 1. Components of 5 is used. Figure 27 and 28 are used to determine which k clusters to choose for dataset 1. Figure 27 shows that it continuously decreases and slows down around cluster 6. For silhouette score, the average is 0.2406. (Max: cluster 2 with 0.3685, Min: cluster 10 with 0.1896) Silhouette score after applying K-mean decreased from 0.5306 to 0.3685. Cluster 6 is the optimal cluster by looking at two graphs. For GMM, tied covariance type had the highest average silhouette score

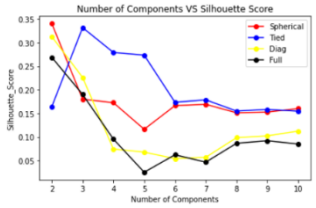

Figure 29: # of Components VS Silhouette

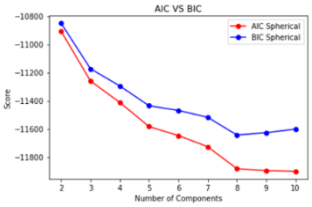

Figure 30: # of Components VS Score

compare to other types. But as number of components increased, spherical covariance type caught up and gets higher at component 10. Silhouette score of spherical covariance type is 0.1789. (Max: cluster 2 with 0.3403, Min: cluster 10 with 0.1605) Optimal cluster to use by looking at the silhouette score is 6 or 7. Compare to the original cluster (without dimentionality reduction), there is a significant drop of the score. By looking at Figure 30, Component 8 had the lowest score for BIC and 10 had the lowest for AIC. This analysis weight BIC more since BIC has more penalty or restrictions that is applied. Optimal component in this case is 8.

K-mean Clustering and GMM with ICA (Dataset 2):

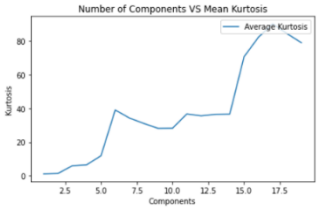

Figure 26: # of Clusters VS Kurtosis

Figure 26 shows the kurtosis mean graph that is used in ICA for dataset 2. The curve increases continuously and reach by the time it reaches component 16, kurtosis hits the maximum. Then, it drops after component 16. By looking at the graph, component 5 is optimal. We do not want to use really low number of components because it means the data has not been separated.

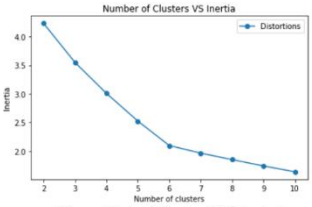

Figure 31: # of Clusters VS Kurtosis

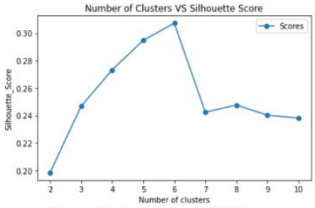

Figure 32: # of Clusters VS Silhouette

K-mean clustering and GMM is done after applying ICA algorithm for dataset 2. Figure 31 and 32 are used to determine which k clusters to choose. Figure 31 shows that it continuously decreases and slows down around cluster 6. For silhouette score, the average is 0.2543. Unlike other graphs, cluster 6 had the maximum silhouette score which means when there are 6 clusters, data are close enough to the centroid compare to other number of clusters.

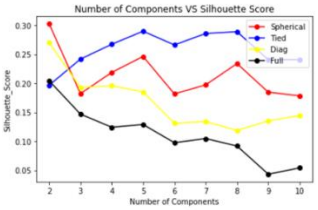

Figure 33: # of Components VS Silhouette

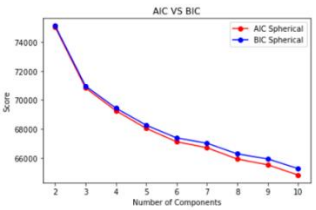

Figure 34: # of Components VS Score

For GMM, tied covariance type had the higher silhouette score compare to the other types followed by spherical. Spherical type will be used for thee analysis to compare the result with K-means. The average silhouette score is 0.2139 where cluster 2, 5, 8 are above the average. Figure 34 is BIC and AIC graph where it keeps on decreasing as number of components increases. For this case, there seems to be a silhouette score gap between cluster 8 and 9, 10. So the optimal number of clusters to use is 8.

Clustering with Dimensionality reduction algorithms (RP):

Random projection (RP) provides an efficient way to reduce dimensionality. It has relatively low accuracy since it is using randomly generated matrix but has faster processing times. There are two main random projections which are Gaussian random projection and sparse random projection. For this analysis, Gaussian random projection is used. Compare to PCA or SVD, random projection is very cheap to compute. Also, it is data independent.

K-mean Clustering and GMM with RP (Dataset 1):

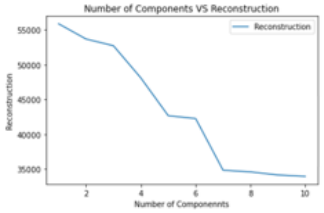

Figure 35: # of components vs reconstruction

Figure 34 is the reconstruction error graph calculated by doing the inverse of dataset 1. Component range of 1 to 10 is used to see which component has the fewest error. It continuously decreases and when it reaches component 7, the rate slows down. Components from 7 up to 10 have the lowest reconstruction errors.

K-mean clustering and GMM is done after applying RP algorithm for dataset 1. Figure 36 and 37 are used to determine which k clusters to choose. Figure 36 shows that it continuously decreases and slows down around cluster 4. For silhouette score, the average is 0.5488. Silhouette score was higher than expected. Cluster 4 is reasonable since it is above the average silhouette score and good amount of distance between each inertias.

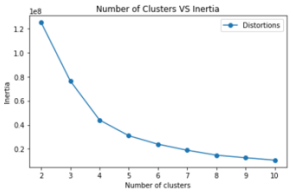

Figure 36: # of Clusters VS Kurtosis

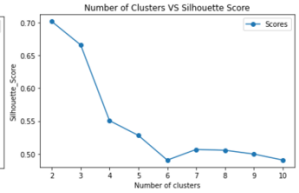

Figure 37: # of Clusters VS Silhouette

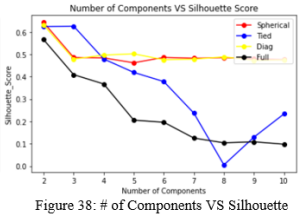

Figure 38: # of Components VS Silhouette

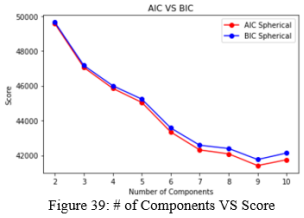

Figure 39: # of Components VS Score

For GMM, diag and spherical covariance type had the higher silhouette score compare to the other types. The average silhouette score for spherical covariance is 0.5011. Except for component 2, all the other components have similar silhouette scores. Figure 39 is BIC and AIC graph where it keeps on decreasing as number of components increases but bounce back after component 9. Clusters of 9 is optimal for this case. Compare to K-mean clustering, GMM had lower silhouette scores.

K-mean Clustering and GMM with RP (Dataset 2):

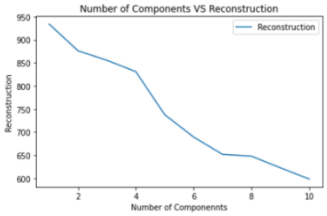

Figure 40: # of components vs reconstruction

Like dataset 1, reconstruction error keeps decreasing as component size decreases. Component 10 which has the lowest reconstruction error is chosen as optimal.

K-mean clustering and GMM is done after applying RP algorithm for dataset 2. Figure 41 and 42 are used to determine which k clusters to use. Figure 41 shows that it continuously decreases and slows down around cluster 4. For figure 42 (silhouette score), the average is 0.3654. Compare to ICA and PCA, RP had higher silhouette average. Cluster 4 is reasonable since it is above the average.

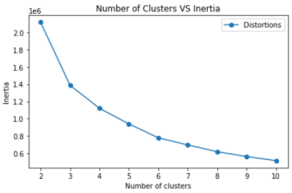

Figure 41: # of Clusters VS Kurtosis

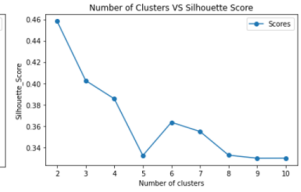

Figure 42: # of Clusters VS Silhouette

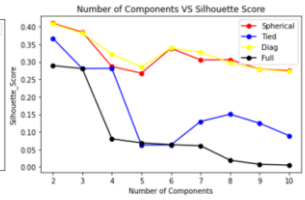

Figure 43: # of Components VS Silhouette

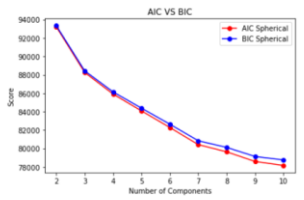

Figure 44: # of Components VS Score

For GMM, diag and spherical covariance type had the higher silhouette score compare to the other types. The result was just like k-means clustering. The average silhouette score for spherical covariance is 0.3171. component 3 and 6 has high silhouette score according to figure

43. Figure 44 shows a BIC and AIC graph where it keeps on decreasing as number of components increases.

Clustering with Dimensionality reduction algorithms (TruncatedSVD):

Lastly, truncated SVD is used. It is similar method with PCA but has significant difference where SVD does not center the data before computing eigenvalue.

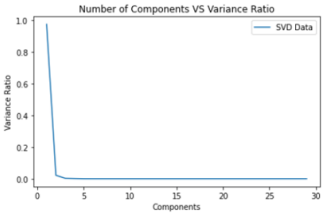

Figure 45: # of Components VS Variance

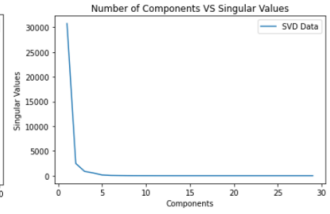

Figure 46: # of Components VS Singular

Like PCA, variance and eigenvalue are calculated and shown on the graph. Figure 45 shows that the variance ratio reaches 0 when there are 5 components. (In other words, ratio is equal to 1.00) Figure 46 also shows a decrease of eigenvalue reaching zero around component 5.

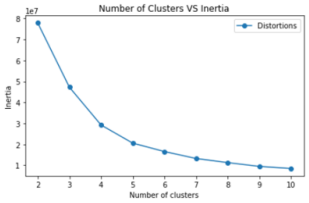

Figure 47: # of Clusters VS Inertia

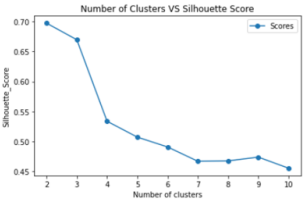

Figure 48: # of Clusters VS Silhouette

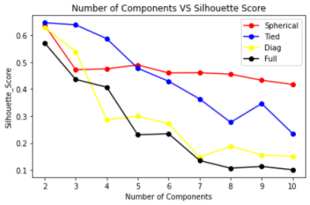

Figure 49: # of Components VS Silhouette

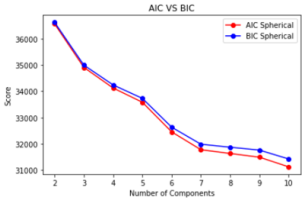

Figure 50: # of Components VS Score

K-mean clustering and GMM is done after applying TruncatedSVD algorithm for dataset 1. By looking at Figure 47 and 48, cluster 4 is optimal since it has decent score for both inertia and silhouette scores. Average score was 0.5290 which is like our original score before reduction. For GMM, spherical covariance type had the higher silhouette score compare to the other types. The average silhouette score for spherical covariance is 0.4780. Component 5 to 10 have similar silhouette scores. Figure 44 shows a BIC and AIC graph where it keeps on decreasing as number of components increases. Optimal component is between 7 to 10.

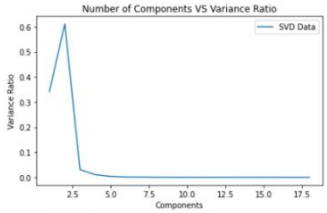

Figure 51: # of Components VS Variance

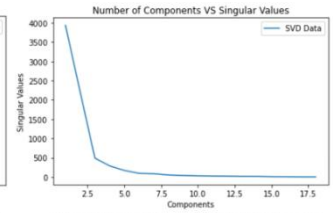

Figure 52: # of Components VS Singular

Figure 51 and 52 is the variance and eigenvalue for dataset 2. Variance hit the max at component 2 and drops. Variance reaches 0 when there are 5 components. Eigenvalue drops dramatically until number of components hit 7. Optimal component of 7 is used for future analysis.

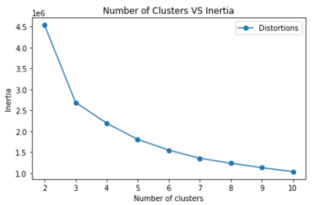

Figure 53: # of Clusters VS Inertia

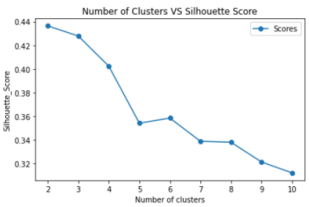

Figure 54: # of Clusters VS Silhouette

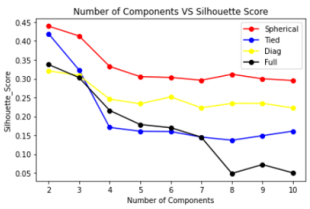

Figure 55: # of Components VS Silhouette

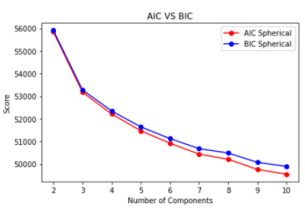

Figure 56: # of Components VS Score

K-mean clustering and GMM is done after applying TruncatedSVD algorithm for dataset 2. By looking at Figure 53 and 54, cluster 4 is optimal since it has high score for both inertia and silhouette scores. Average score was 0.0.3623. For GMM, spherical covariance type had the

highest silhouette score compare to the other types. The average silhouette score for spherical covariance is 0.0.3329. Component 5 to 10 have similar silhouette scores. Figure 44 shows a BIC and AIC graph where it keeps on decreasing as number of components increases. Optimal component to use here is 8.

Dimensionality reduction algorithms to Neural Network:

In this section, all 4 dimensionality reduction algorithms will be applied to neural network and results will be compared. Dataset 1 is used for this experiment. From assignment 1, Neural network had accuracy of 0.92 with 10 false positives and 13 false negatives on confusion matrix. After hyperparameter tuning, accuracy went up to 0.93 with 6 false positives and 10 false negatives. Cross validation score was 0.93 and it took 3.232 seconds to run. This data will be a benchmark to compare with 4 different dimensionality reduction algorithms described above.

	Original	PCA	ICA	RP	SVD
False Positive	10	10	4	13	8
False Negative	13	10	13	16	16
Precision	0.903	0.906	0.959	0.874	0.918
Recall	0.877	0.906	0.877	0.849	0.849
F1-Score	0.890	0.906	0.916	0.861	0.882
Accuracy	0.919	0.930	0.940	0.898	0.916
Cross Validation	0.905	0.944	0.954	0.926	0.940
False Positive	6	13	5	6	7
False Negative	10	6	14	20	15
Precision	0.921	0.885	0.948	0.935	0.929
Recall	0.877	0.943	0.868	0.811	0.858
F1-Score	0.899	0.913	0.906	0.869	0.892
Accuracy	0.926	0.933	0.933	0.909	0.923
Cross Validation	**0.947**	**0.954**	**0.958**	**0.933**	**0.922**
Time	**3.232**	**2.299**	**2.970**	**0.706**	**3.126**

Every dimensionality reduction algorithm is applied to the dataset and base on that dataset, neural network learner is experimented. Compare to the original benchmark, time is reduced. It took 3.232 seconds when the data is not reduced but all the other datasets after reduction reduced the time. Especially, time it took to train random projection was very fast (0.706 seconds). My hypothesis was “Reduction algorithms trade accuracy with time. If time reduces, accuracy score will go down.” But the hypothesis was not always correct since some of the algorithms had better cross validation score and accuracy compare to the original benchmark. Original cross validation score was 0.947. PCA and ICA exceeded the benchmark but RP and SVD could not. For RP, some accuracy is lost since the matrix is randomly generated and at the same time, the cost function reduced a lot. It is still a good sign that every dataset after the reduction achieved accuracy greater than 0.90.

One possibility that the result is not straightforward is because data is not huge enough. There are around 600 rows with 30 attributes. One assumption that I could make is that there were not much of a data to reduce by those algorithms to see the clear difference between the original benchmarks. But still, the cost or time to compute decreased without losing too much of accuracy. This applies to all the other scores such as precision, recall and f1-score before and after the hyperparameter tuning.

Clustering algorithms to Neural Network:

In this section, K-mean algorithm and GMM algorithm will be applied to neural network. So, total of 8 experiments are done. n_clusters and n_components values that are used for this experiment is decided by the previous section. In this analysis, accuracy and time is again compared. Initially, each reduced dataset is called, and clustering algorithm ran based on the reduced dataset. After that, KM and EM columns were added to the dataset. Most updated dataset is now used in neural network algorithms. Following table is the result after running all the combinations.

	Original	PCA	ICA	RP	SVD
False Positive	10	6	10	20	10
False Negative	13	11	11	11	14
Precision	0.903	0.941	0.905	0.826	0.902
Recall	0.877	0.896	0.896	0.896	0.868
F1-Score	0.890	0.918	0.900	0.960	0.885
Accuracy	0.919	0.940	0.926	0.891	0.916
Cross Validation	0.905	0.961	0.937	0.873	0.922
False Positive	6	6	13	20	12
False Negative	10	15	12	13	17
Precision	0.921	0.938	0.879	0.823	0.881
Recall	0.877	0.858	0.887	0.877	0.840
F1-Score	0.899	0.897	0.883	0.849	0.860
Accuracy	0.926	0.926	0.912	0.884	0.898
Cross Validation	**0.947**	**0.933**	**0.937**	**0.919**	**0.929**
Time	**3.232**	**3.924**	**3.216**	**1.269**	**2.899**

As shown on the table, time run time (cost function) of ICA, RP and SVD was lower compare to the original. PCA took more time compare to the original data and one of the possibilities is that the dataset is not big enough to be reduced which makes original dataset competitive enough. But overall, the function ran fast. Cross validation accuracy decreased. This experiment can be seen successful since all the cross-validation score is above 0.90. PCA had 0.933, ICA 0.937, RP 0.919 and SVD 0.929. This applies to all the other scores such as precision, recall, f1-score before and after hyperparameter tuning.

In conclusion, both dimensionality reduction algorithms and clustering algorithm trade accuracy with the cost (time). But important thing is that the cost decreased a lot, but accuracy did not. Some experiment showed even higher accuracy compare to the benchmark. These techniques will be very helpful when there is a huge dataset, and we do not want it to run forever.

Citation:

"6.6. Random Projection¶." *Scikit*, scikit-learn.org/stable/modules/random_projection.html.

"Sklearn.decomposition.FastICA¶." *Scikit*, scikit-learn.org/stable/modules/generated/sklearn.decomposition.FastICA.html.

"Sklearn.decomposition.PCA¶." *Scikit*, scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html.

"Sklearn.decomposition.TruncatedSVD¶." *Scikit*, scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html?highlight=truncatedsvd.