

# OMSCS\_CS7641 HW3

## Unsupervised Learning and Dimensionality Reduction

Clement Li (cli620) Fall 2020

### Introduction:

The purpose of this project is to explore unsupervised learning algorithms and dimensionality reduction techniques. Unsupervised learning algorithms include k-mean clustering and expected maximization (EM). Dimensionality reduction algorithms include Principal Component Analysis (PCA), Independent Component Analysis (ICA), Randomized Projections (RP) and one feature selection algorithm of our choosing (KPCA). The goal of this project is to compare these algorithms.

There are 7 parts to this paper. 1) we will describe the datasets 2) take a look at how we will evaluate our clustering/ dimension reducing results 3) apply k-means clustering and expected maximization on two sets of data 4) apply the dimensionality algorithms 5) reapply the two clustering algorithms after the four dimensionality reduction algorithms 6) train a neural net on the dim-reduced data sets and 7) retrain the neural net with data post-processed with the cluster algorithm on top of the dim-reduce algorithms. The code will use the sk-learn libraries for GaussianMixture (EM), KMeans, PCA, ICA, RP and KPCA. I modified ezerilli's API for my own experiments and results. [1]

### About Data Sets

The datasets used is an early stage diabetes risk and prediction data set obtained from the UC Irving machine learning repository [2] and the default sklearn library breast-cancer dataset [3].

This early diabetes detection dataset, pulled down from the UC Irving machine learning repository, has each row represents an individual patient. Each row describes whether they have any of the 16 characteristics of diabetes and a flag of if they were diagnosed with diabetes. Some example characteristics includes age, gender, obese, etc. There are 521 patients in this dataset. The data is mostly binary besides the age field. We used this dataset in homeworks 1 and 2.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Age	Gender	Polyuria	Polydipsia	sudden w	weakness	Polyphagia	Genital th	visual blui	itching	Irritability	delayed h	partial pai	muscle sti	Alopecia	Obesity	class
2	40	Male	No	Yes	No	Yes	No	No	No	Yes	No	Yes	No	Yes	Yes	Yes	Positive
3	58	Male	No	No	No	Yes	No	No	Yes	No	No	No	Yes	No	Yes	No	Positive
4	41	Male	Yes	No	No	Yes	Yes	No	No	Yes	No	Yes	No	Yes	Yes	No	Positive
5	45	Male	No	No	Yes	Yes	Yes	Yes	No	Yes	No	Yes	No	No	No	No	Positive
6	60	Male	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Positive
7	55	Male	Yes	Yes	No	Yes	Yes	No	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Positive

The breast cancer data set is a binary classification dataset with 569 samples total and 30 features. This dataset is similar to the diabetes dataset but with more features and non-binary features. The values are detailed in the following table:

Param	Min	Max	Param	Min	Max	Param	Min	Max	Param	Min	Max
radius (mean)	6.981	28.11	radius (std err):	0.112	2.873	radius (worst):	7.93	36.04	concave points (mean):	0.0	0.201
texture (mean)	9.71	39.28	texture (std err):	0.36	4.885	texture (worst):	12.02	49.54	symmetry (mean):	0.106	0.304
perimeter (mean)	43.79	188.5	perimeter (std err):	0.757	21.98	perimeter (worst):	50.41	251.2	fractal dim (mean):	0.05	0.097
area (mean):	143.5	2501.0	area (std err):	6.802	542.2	area (worst):	185.2	4254.0	concave points (std err):	0.0	0.053
smoothness (mean):	0.053	0.163	smoothness (std err):	0.002	0.031	smoothness (worst):	0.071	0.223	symmetry (std err):	0.008	0.079
compactness (mean):	0.019	0.345	compactness (std err):	0.002	0.135	compactness (worst):	0.027	1.058	fractal dim (std err):	0.001	0.03
concavity (mean):	0.0	0.427	concavity (std err):	0.0	0.396	concavity (worst):	0.0	1.252	concave points (worst):	0.0	0.291
fractal dim (worst):	0.055	0.208	symmetry (worst):	0.156	0.664						

## Analysis Techniques quick summary

### Akaike Information Criterion (AIC) / Bayesian Information Criterion (BIC)

Akaike and Bayesian Information Criterion are two ways of scoring a model based on its log-likelihood and complexity. Respectively, the formulas for these criterion are:  $AIC = -2/N * LL + 2 * k/N$  and  $BIC = -2 * LL + \log(N) * k$ . (LL = log-likelihood, N = number of examples, k = number of parameters). These values are to be minimized for both criterions. [4] We will use this mainly to explore Expected Maximization algorithm to find the parameters (components and covariance type) with the highest log likelihood.

### Silhouettes Analysis

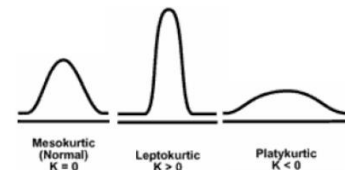
Silhouettes analysis can be used to study the separation distance between the resulting clusters. This measure has a range of [-1, 1] measuring how close each point in one cluster is to points of neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary. Negative values indicate those samples are assigned to the wrong cluster. The thickness shows the cluster sizes. [5] We will use this mostly to assess K-mean clustering to see how well the clusters inter-fit.

### Variance / Cumulative Variance

Variance refers to the sensitivity of the learning algorithm to the specifics of the training data, e.g. the noise and specific observations. [6] Variance is mostly used for PCA and KPCA to measure the variance in each principal component. The sum of the sample variances of all individual variables is called the *cumulative variance*. PCA is designed to maximize the first k components and minimize the variance of the last p-k components. We try to choose k big enough to make the lost information sufficiently small. [7] Hence, the k at which the cumulative variance converges, is the k we want. This will minimize the amount of features but still retain covering all the data.

### Kurtosis

Kurtosis is the measure of peakedness or flatness of a distribution. See image on the right. We want K to be greater than 0, particularly as high as possible. This indicates that the values in that independent component is more centralized and far away from the other components as possible. If the values are Platykurtic ( $k < 0$ ) then it is more likely to spill into its neighbors. [8]

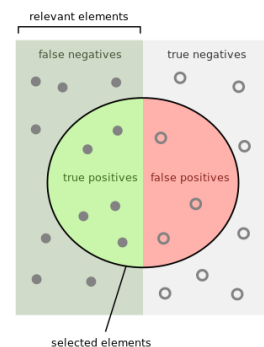


### Mean Square Error (MSE) for the Reconstruction

To evaluate randomized projections, we will compare the MSE between the original data and the reconstructed data, which is the reduce feature data space weights cross the labels added to the new average data. This error should be minimized because we do not want to lose data in the components we got rid of.

### t-SNE and PCA

To view the cluster, we visualized it in two ways: (1) via PCA where it plots out the first vs second principal component and color each of the different types of classes with a different color (2) via T-Distributed Stochastic Neighboring Entities (t-SNE) where it plots out the first and second components of the algorithm specifically reduced down to two components. [9] This ends up with two good visuals on how well the respective dimension reduction algorithms separated the data and individual data points that were classified right or wrong. We will also compare the results to the true classes of the data to see how well it did.



### F1 Score/ Confusion Matrix

F1 score is the harmonic mean of the precision values and recall value. Precision is the positive predictive value (true positives/(true positives + false positives)) and recall is the sensitivity value (true positives / (false negatives + true positives)). These two metrics measures 1) how many selected items are relevant and how many relevant items are selected.

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2tp}{tp + \frac{1}{2}(fp + fn)} \quad [10]$$

Confusion Matrix details the amount of true positive, false positive, false negative, true negatives classified (see left) . Ideally, we want True + and True – to be high and False + and False – to be low.

True +	False +
False -	True -

# Clustering

## Results

### K Means

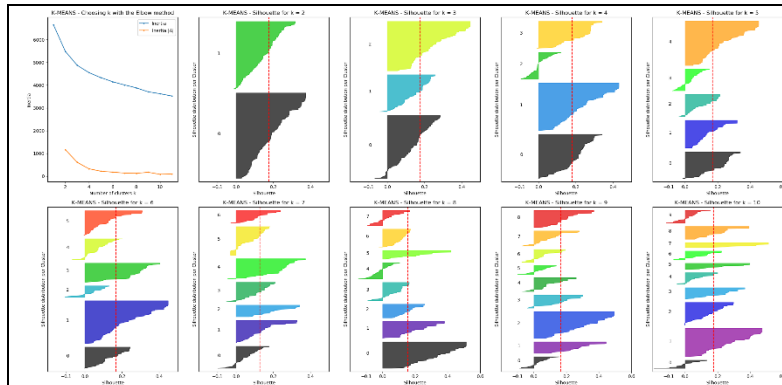


Figure 1: K-means algorithms analysis of Diabetes dataset using a Silhouette technique. Here 2 clusters seem to have the least amount of errors indicated by the negative values. This makes sense because the dataset should only classify to positive or negative diabetes predictions.

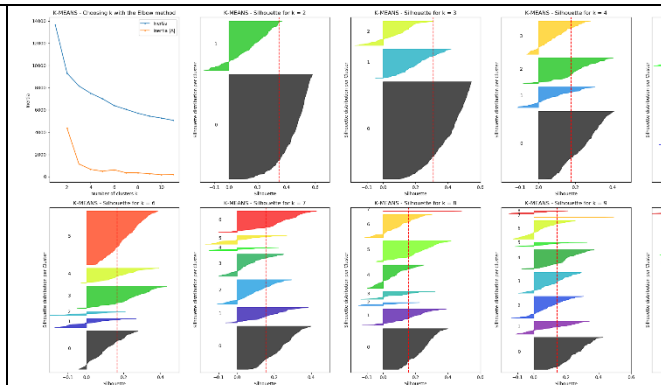


Figure 2: K-means algorithms analysis of Breast Cancer dataset using a Silhouette technique. There are not any good cluster sizes. Cluster of 2 has the least amount of errors in the negatives. All other clusters only spread out the error into the new clusters. With a cluster of 4 as an example, the error was spread out between the first and second clusters. This is not necessarily a good thing because though the amount of data has shrunk relatively to the 2-cluster results, relative to its own cluster the percentage of wrong is still just as large.

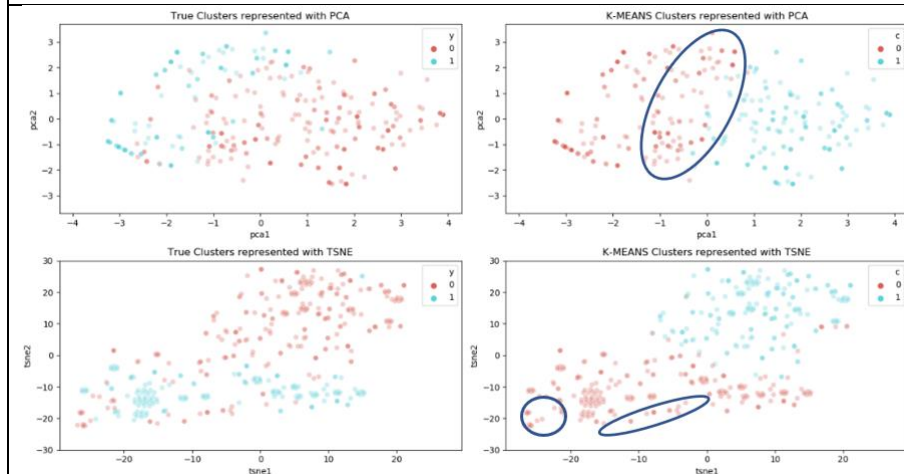


Figure 3: K-means clustering with 2 clusters on the Diabetes dataset. Using the clusters found via silhouetting, we see some errors in the clusters compared to the true clusters mostly in the boundaries when viewing with PCA. This is highlighted with the blue circle. This indicates that KMeans has challenges giving well defined boundaries. When viewing with TSNE, we see that most of the values are correctly classified. However, outlier samples (highlighted by the blue circles) that are embedded in the opposite clusters are easily mislabeled. It is also interesting to note that the classes are swapped. This is a characteristic of the algorithm where it randomly chooses two points and find the means around it until it converges. This does not label those two clusters during the process. It will be up to the user's interpretation or post-analysis likeness work to match them up with a label.

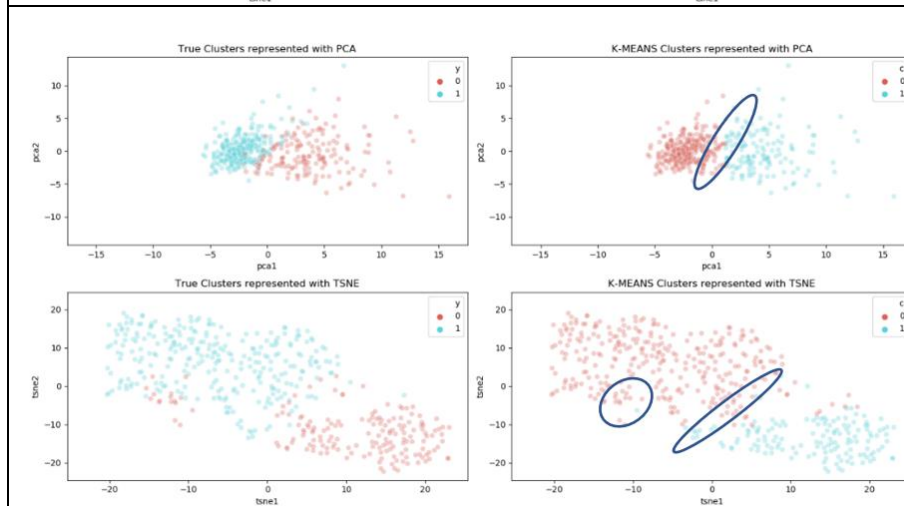


Figure 2: K-means clustering with 2 clusters on the Breast Cancer dataset. These values show that the clusters are well grouped. The PCA has better alignment than with the diabetes dataset but the boundaries still have some errors. In the TSNE view, there is the similar boundary issue but also a small cluster that did not manage to be labeled correctly. This shows that if outlier data points are littered in the opposite label territory, the KMeans algorithm will have difficulties labeling it correctly.

## Expected Maximization (EM)

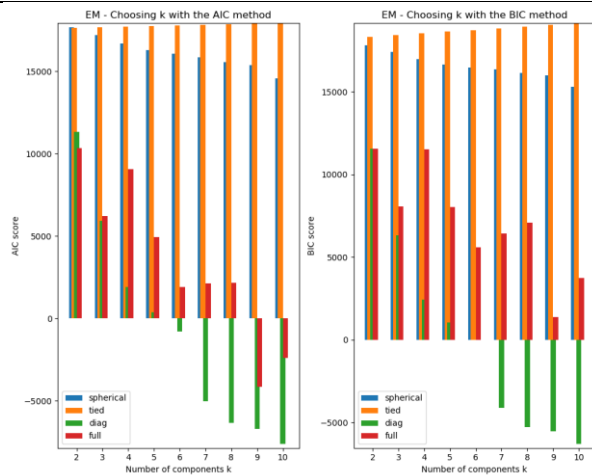


Figure 5: AIC/BIC evaluation for diabetes dataset clusters. Here the lowest criterion values for both AIC and BIC is using 10 components and diag covariance. It is interesting to note that AIC and BIC come in both negative and positive values. From the analysis technique quick summary section, this shows that negative values either means the likelihood is large or the number of components is small. From looking at our data, the number of components is medium sized this would indicate that anything more than negative would indicate undesired likelihood.

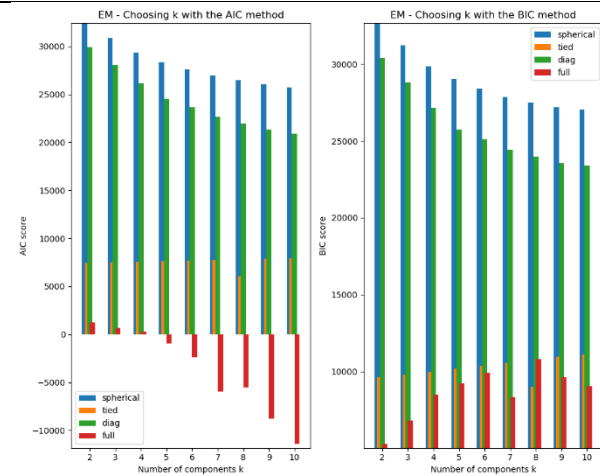


Figure 6: AIC/BIC evaluation for breast cancer dataset clusters. The lowest criterion value for AIC is 10 component and full covariance. However, 2 component and full has the lowest values for using BIC. Since this is a smaller dataset with fewer features, BIC is more likely to choose models that are too simple [4]. In addition, following our logic in the previous figure caption, we would not want 2 components and full covariance because the AIC value is positive. This would indicate low likelihood; therefore, we will be using 10 components and full covariance.

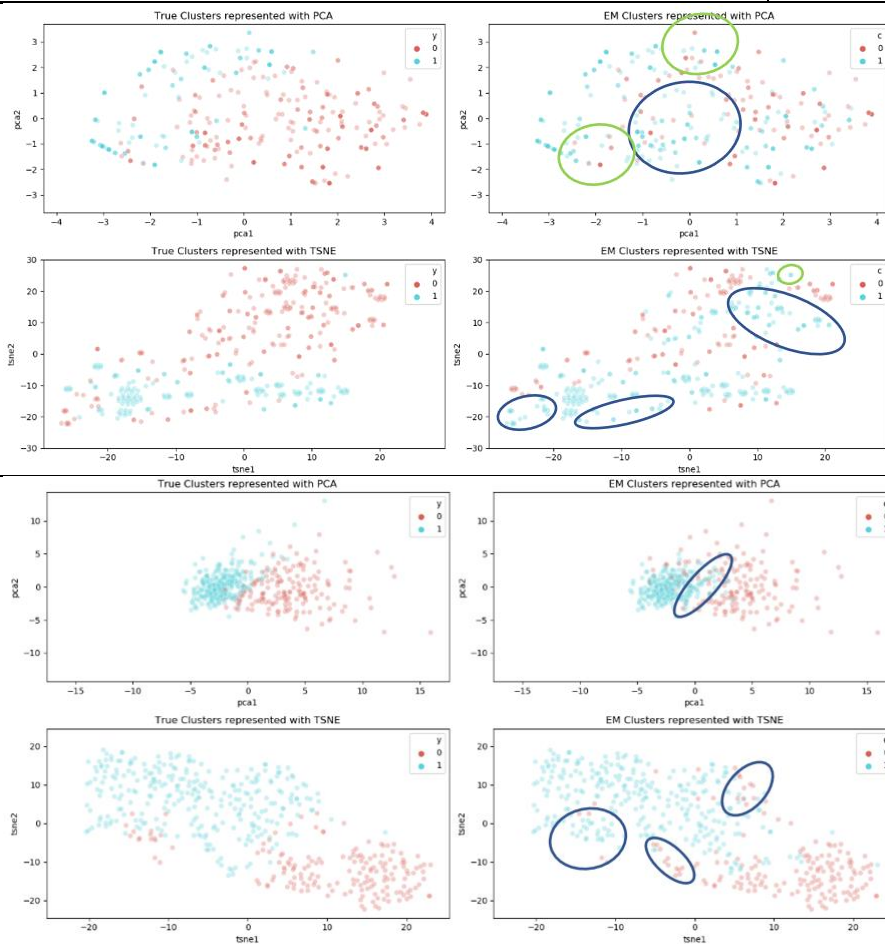


Figure 7: EM clustering with 2 clusters on the diabetes dataset. The clustering is much more mixed up here due to the characteristics of the algorithm. In the PCA and TSNE plot, the algorithm doesn't look for clear geometric characteristics but through statistical association within its features. Though it does not give a clean classification visualization, it is able to correctly classify some datapoints (in green circles) embedded that KMeans have missed. However, the trade off is there are more obvious clusters that were missed such as the ones in the blue circles.

Figure 8: EM clustering with 2 clusters on the Breast Cancer dataset. In both plots, the clusters are well grouped. The analysis for the breast cancer dataset is like the one for diabetes dataset. It is interesting to note that EM was still unable to correctly classify that group of datapoints in the left circle in the TSNE plot.

Analysis

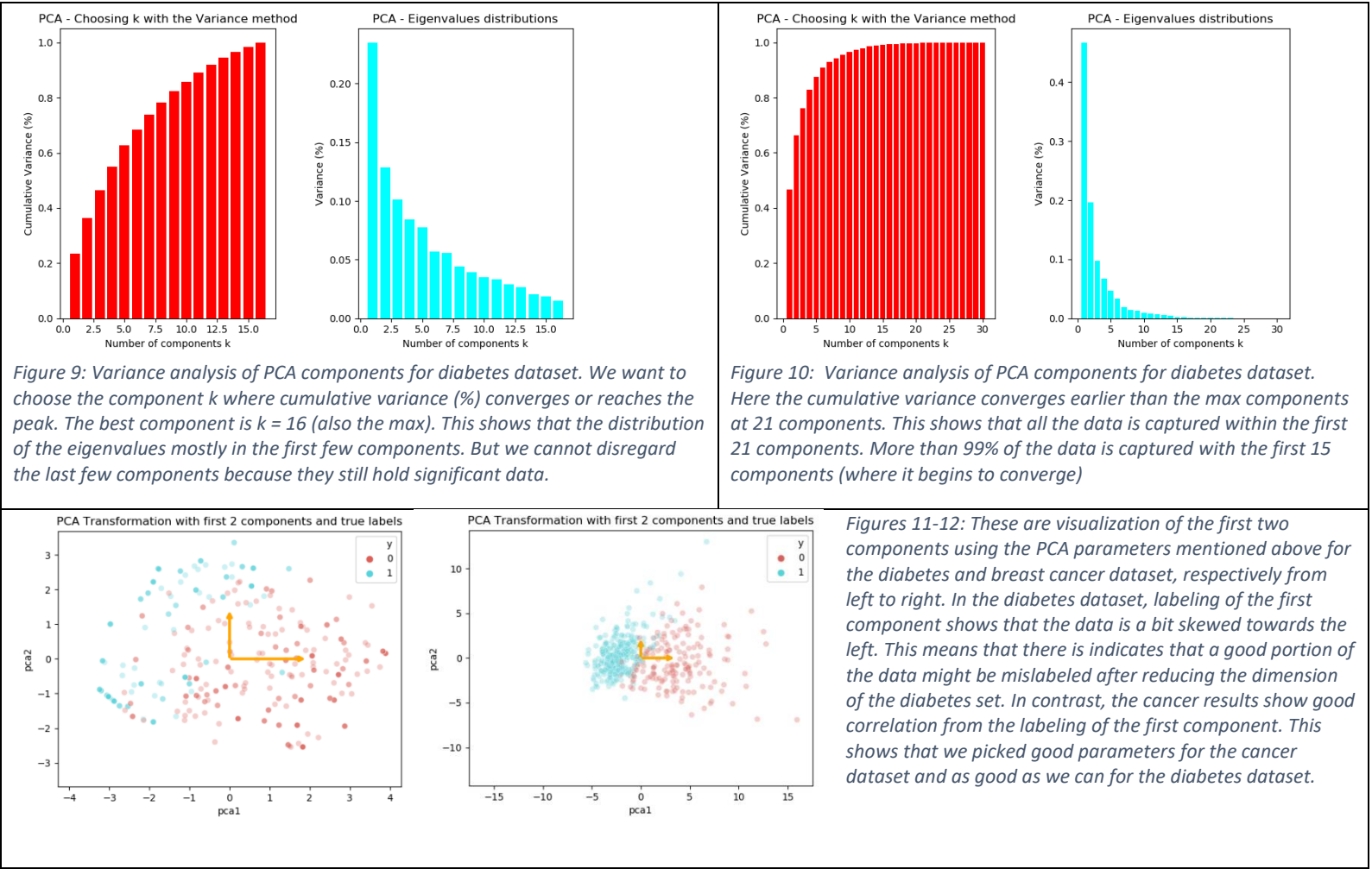
In addition to the observations listed in the Figure captions, here is a table of summarized pros/cons of each algorithm.

	Pros	Cons
Kmeans	Able to make good clear clusters	Can miss outliers hidden in the clusters Clusters doesn't directly reflect classes Boundaries can be unclear
EM	Able to pick out some outliers missed by KMeans	Classifications are sparse and interweave into other clusters. Boundaries can be unclear

Dimension Reduction on Datasets

Results / Analysis

PCA





## ICA

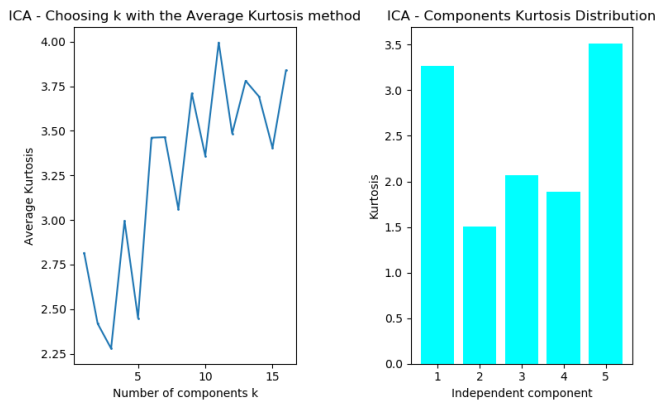


Figure 13: Analysis of kurtosis of with varying components for diabetes data. Here we show that at components of 11, we have the highest average kurtosis and where the curve starts to flatten out.

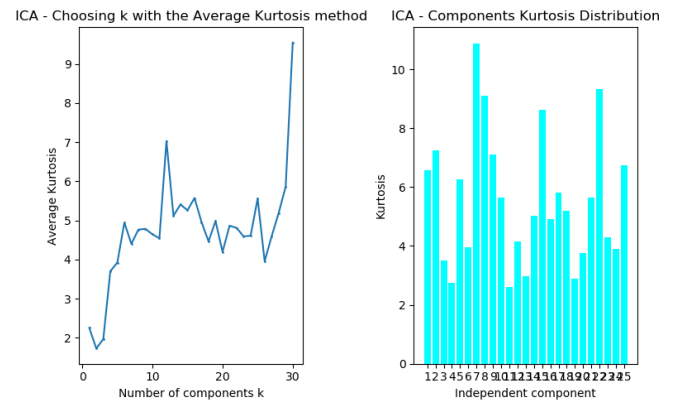
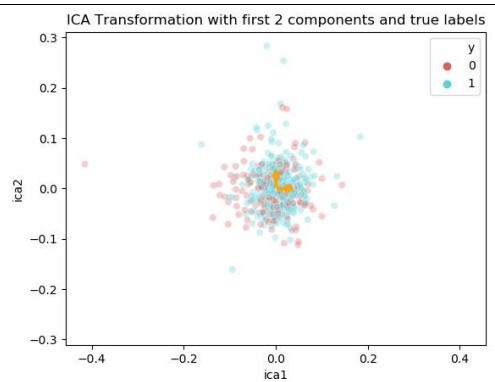
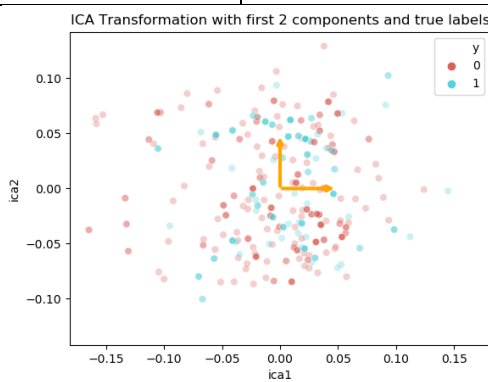


Figure 14: Analysis of kurtosis of with varying components for cancer data. The average kurtosis has two peaks, at 12 and at 30. I chose 12 because a reason why kurtosis at 30 is highest is because it indicates that there are many compacted peaks. This seems indicates overfitting.

Figure 15-16: These are visualization of the first two components using the ICA parameters mentioned above for the diabetes and breast cancer dataset, respectively from left to right. The datapoints in both components very jumbled. This indicates that the reduced dataset will predict the true labels poorly. It is important to note that the axis have small range. This shows the parameters obtained what we want with kurtosis in figures 13 and 14; The components are tightly packed together but still on top of each other. This shows that ICA is not a good technique for these two datasets. ICA is good for datasets the two labels are statistically independent of each other. This shows that the two labels in both sets have dependencies in the other label.



## RP

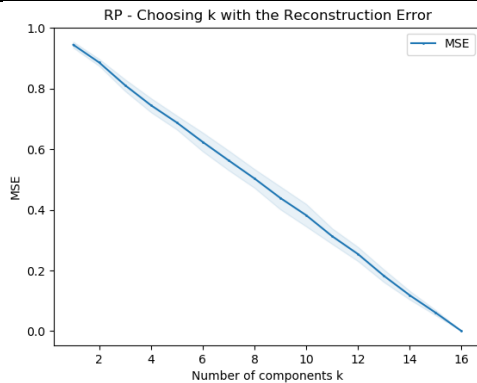


Figure 17: Reconstruction vs original MSE per # of component used for diabetes data. The MSE went down to 0 when we use all 16 components. There were very little variations over all the times it was re-ran. This is likely because the dimensions are small.

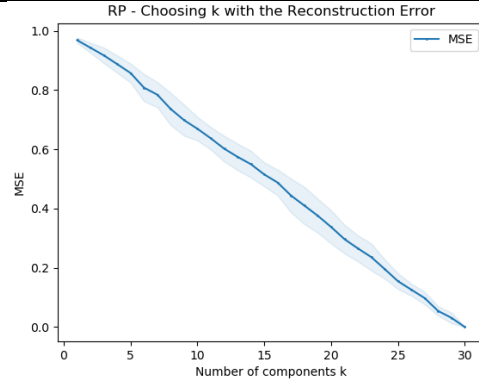
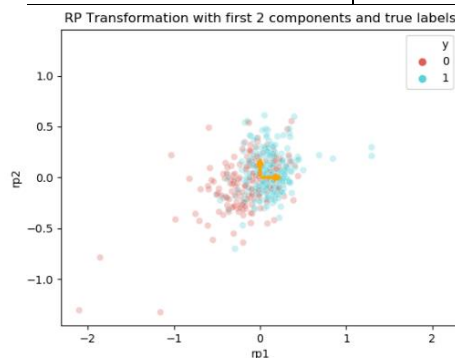


Figure 18: Reconstruction vs original MSE per # of component used for cancer data. Same as the diabetes dataset, the reconstructed data had 0 MSE at 30 components. The variations per run is higher due to having twice the dimensions.



Figures 19-20: These are visualization of the first two components using the RP parameters mentioned above for the diabetes and breast cancer dataset, respectively from left to right. The results show poor separation of the labels for the diabetes dataset. The results for the cancer dataset are better but still have some errors due to how close the two clusters are. An explanation for the poor results is that the first two components is not representative to the full dimension dataset. This is noted in Figure 17 and 18 where the first two components have high MSE.

## KPCA

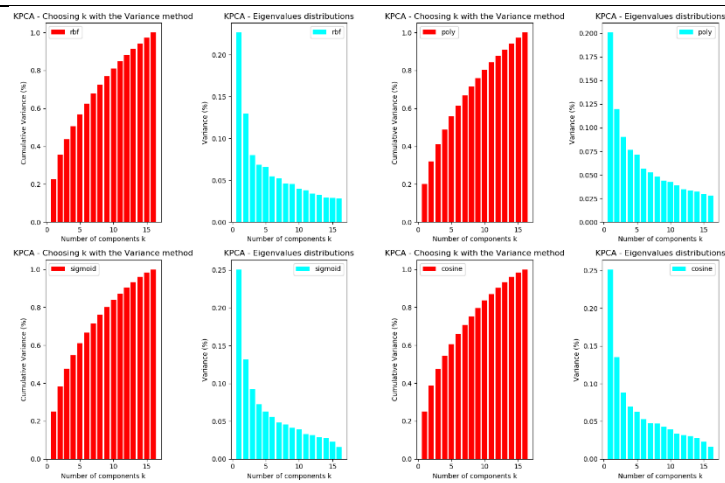


Figure 21: AIC/BIC evaluation for diabetes dataset clusters. In these variances, none of the parameters converges. However, it shows that 16 components with sigmoid or cosine ends at the lowest variance on the last component. Similar to PCA, this shows that the distribution of the eigenvalues mostly in the first few components. But we cannot disregard the last few components because they still hold significant data. Therefore, we use 16 components and sigmoid for this data.

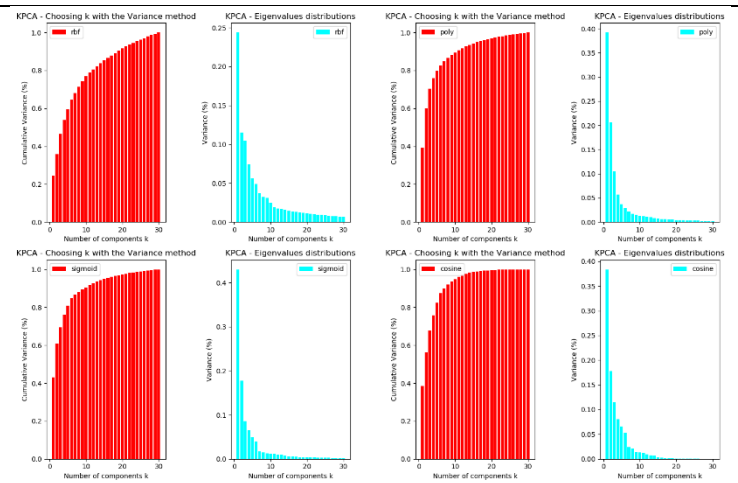
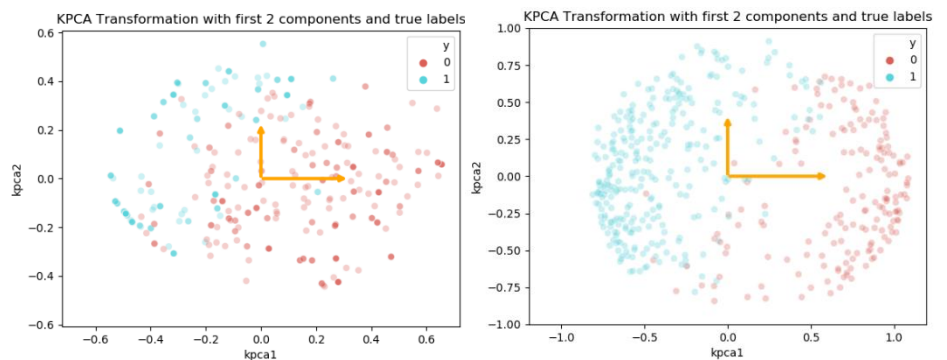


Figure 22: AIC/BIC evaluation for breast cancer dataset clusters. Here we see some convergences while using the cosine kernel. The cumulative variance for the cosine plot converges at 1.0 at around 21 components. This shows that all the data is captured within the first 21 components. More than 99% of the data is captured with the first 15 components (where it begins to converge) We will use 21 components and cosine kernel.

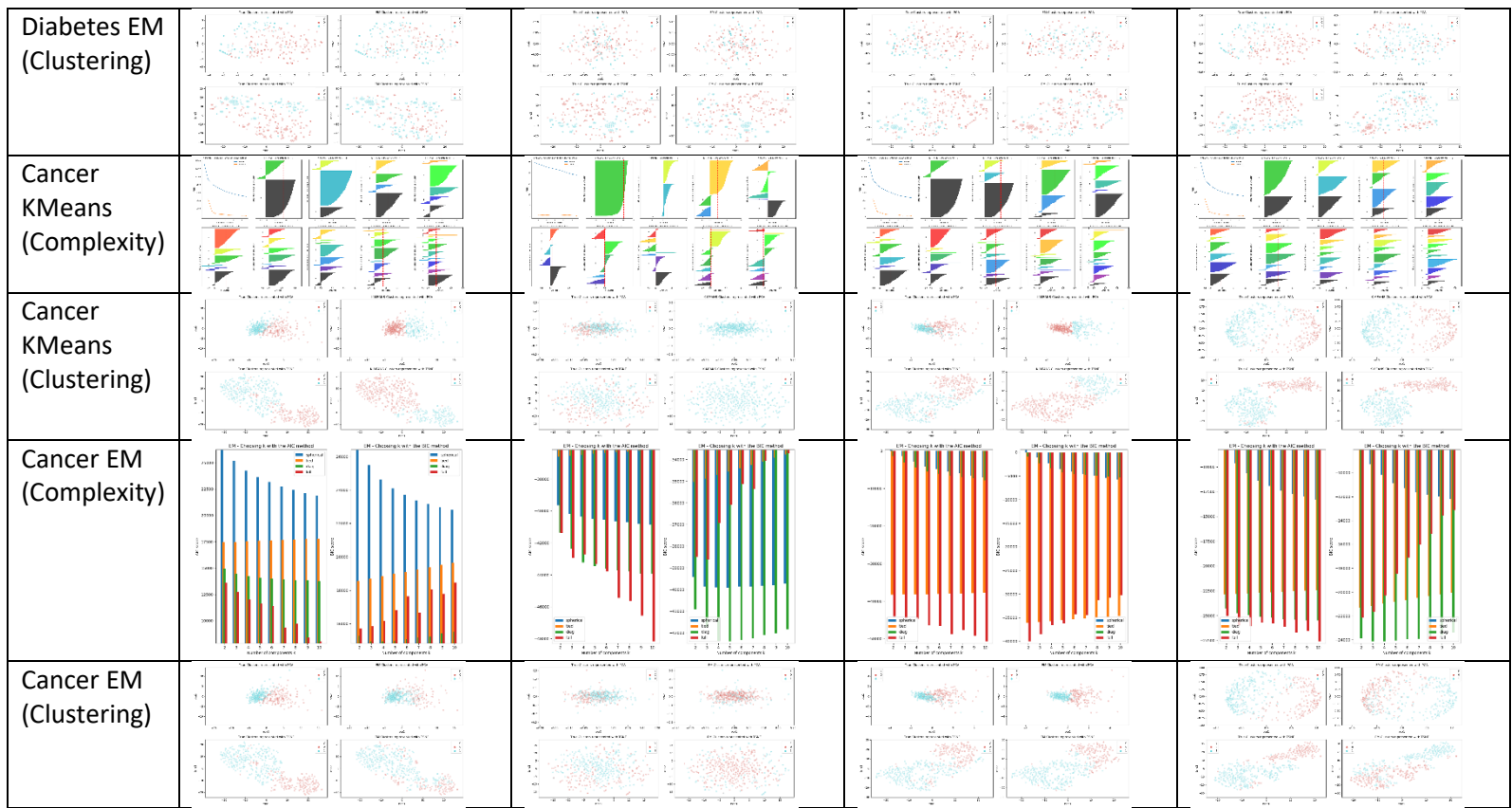
Figure 23-24: These are visualization of the first two components using the KPCA parameters mentioned above for the diabetes and breast cancer dataset, respectively from left to right. The PCA results are very similar to KPCA components, which makes sense because the distribution of the components in Figure 21 for the sigmoid kernel are the same as Figure 9. Though this is true, the other kernels have worst performance than the sigmoid kernel. This indicate that using PCA is sufficient for this diabetes dataset. In the cancer dataset, the component visualization is better compared to PCA because there is a wider separation between the two classes. This shows that by limiting the components to 21 instead of the full 30, we have improved the results.



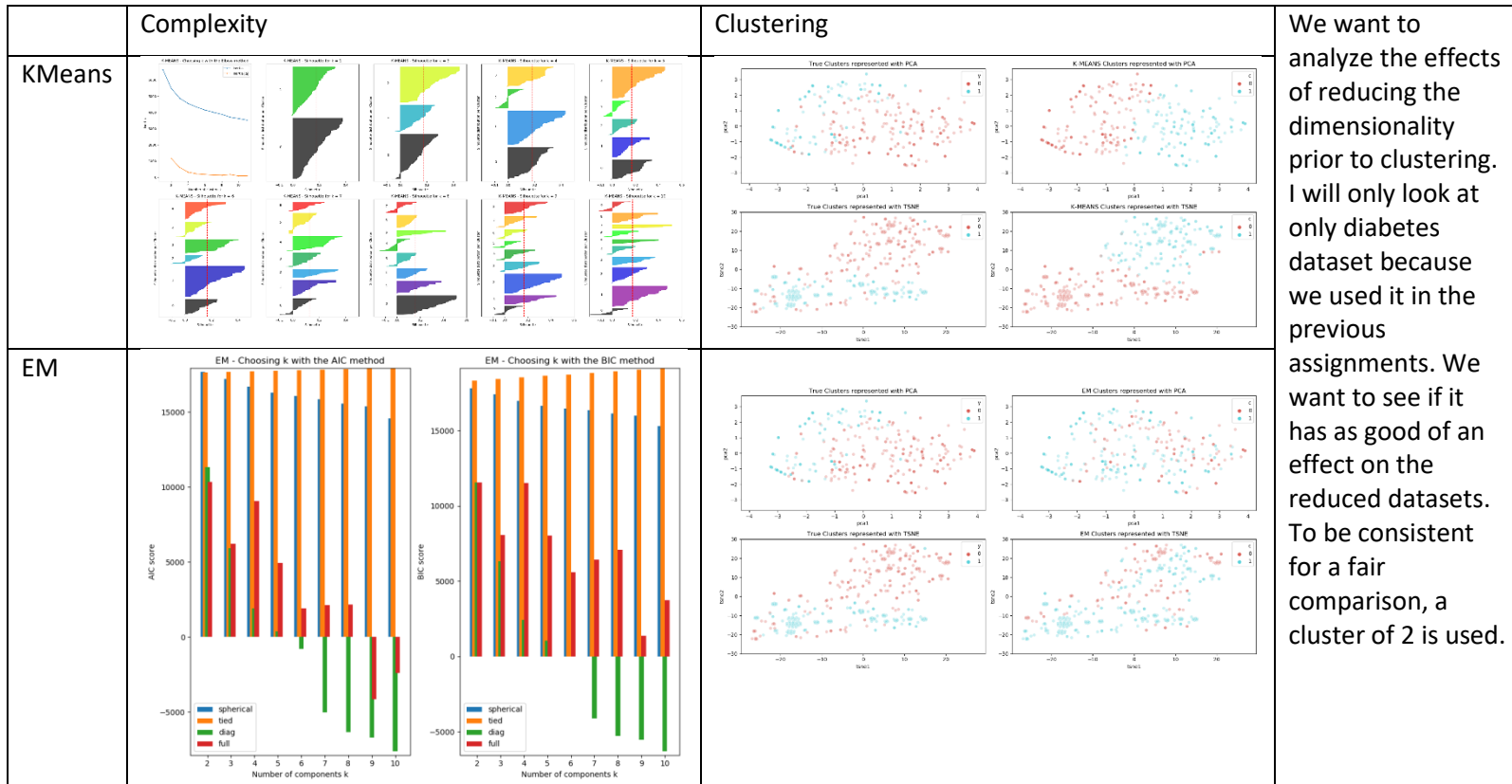
## Clustering on Dimension Reduced Datasets

### Results

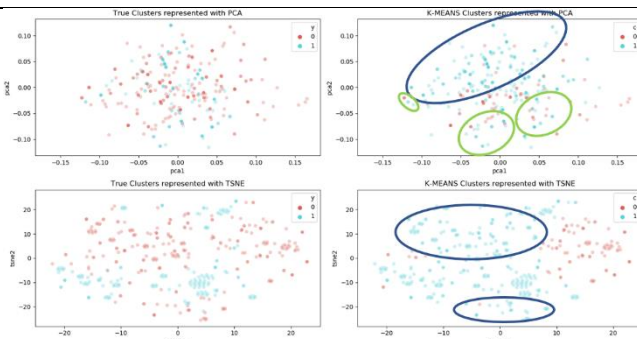
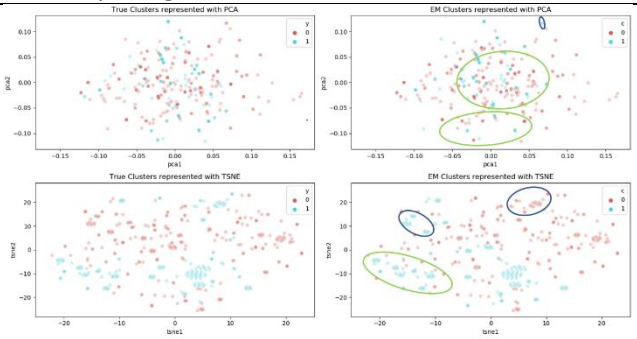
	PCA	ICA	RP	KPCA
Diabetes KMeans (Complexity)				
Diabetes KMeans (Clustering)				
Diabetes EM (Complexity)				



## Analysis





Clusters	PCA	ICA	RP	KPCA
KMeans	Roughly same results as KMeans and EM truth plots. Possibly because we use PCA to view the clusters for both Kmeans and EM. (i.e.: plotting 1 <sup>st</sup> component versus 2 <sup>nd</sup> component.)	 <p>The shape for ICA reduced clusters changed due to the gaussian grouping explained in figure 15. The classification in these clusters show improvement for correctly labeling very mixed true labels. Despite the components being so packed together and not being able to provide a good visualization, KMeans is able to correctly the green circles.</p>	The RP dimension reduced dataset did not have very good clear clusters that KMean can do well on. The performance did not decrease mainly because the original KMeans also did poorly. As mentioned before, the dimensions of these data sets may not be large enough for RP to be optimally used.	The results are similar to PCA and regular KMeans/EM clustering. However, the silhouettes for PCA and KPCA of 2 clusters indicate that more is allocated to the first cluster than the second cluster. This will likely cause more false positives in the classification, since PCA aligned well with KMeans/EM silhouettes. This may be because the sigmoid kernel was used. The additional smoothing might overfit the data if it is largely linear. In the future for better performance, I may use Linear Discriminant Analysis (LDA) because the from this experiment the data appears linear and LDA considers the classes which would make the analysis interesting to compare PCA and LDA.
EM		 <p>EM did a very good job with this ICA reduced dataset. This is likely because EM is a method that measures maximum likelihood which is optimal for an ICA reduced dimensions which fits a tightly clustered gaussian. The remaining data points are statistically best suited in two gaussian peaks. Since we chose a k component that provides the highest average kurtosis, these peaks are as statistically related as possible. This proved to perform well with difficult datapoints such as in the green circles.</p>	Like KMeans, the RP output missed some obvious clusters. As mentioned above, this is likely due to this dataset not being large enough for RP to be optimally used.	

## Neural Net with Dimensionality Reduction for Diabetes Dataset

### Results /Analysis

NN	precision	recall	f1-score	support	Confusion Matrix:	RP + NN	precision	recall	f1-score	support	Confusion Matrix:
0	0.97	1	0.98	64	64 0	0	0.98	0.98	0.98	64	63 1
1	1	0.95	0.97	40	2 38	1	0.97	0.97	0.97	40	1 39
accuracy			0.98	104		accuracy			0.98	104	
macro avg 0.98		0.97	0.98	104		macro avg 0.98		0.98	0.98	104	
weighted 0.98		0.98	0.98	104		weighted 0.98		0.98	0.98	104	
PCA + NN	precision	recall	f1-score	support	Confusion Matrix:	KPCA + NN	precision	recall	f1-score	support	Confusion Matrix:
0	0.98	0.98	0.98	64	63 1	0	0.98	0.84	0.91	64	54 10
1	0.97	0.97	0.97	40	1 39	1	0.8	0.97	0.88	40	1 39
accuracy			0.98	104		accuracy			0.89	104	
macro avg 0.98		0.98	0.98	104		macro avg 0.89		0.91	0.89	104	
weighted 0.98		0.98	0.98	104		weighted 0.91		0.89	0.9	104	
ICA + NN	precision	recall	f1-score	support	Confusion Matrix:						
0	0.9	0.89	0.9	64	57 7						
1	0.83	0.85	0.84	40	6 34						
accuracy			0.88	104							
macro avg 0.87		0.87	0.87	104							
weighted 0.88		0.88	0.88	104							

The first thing to note is that using the same projected data from the previous section, the reduced data proved to be harder to train for the same neural net. The f1-score for all reduced inputs has not exceeded original dataset . Here RP and PCA were able to maintain the f1-score, meaning it is just as reliable as the original dataset results. However, RP and PCA trades more False-Positive for less False-Negatives. This indicates that more of the selected items are relevant and less of the relevant items are selected. This makes sense as we have decreased the dimensions and possibly decorrelating some samples that depended on that omitted feature to be selected correctly. The ICA reduced results is expected because from prior findings, the ICA reduced data need EM clustering to improve its

representation. KPCA should have results similar to PCA but the number of false positives increased significantly. This brings us back to the reasoning in the previous section where more was allocated to the KPCA silhouette for cluster 1 from cluster 2. The explanation is that the sigmoid kernel may have overfitted the data and providing misleading correlations to the neural net.

## Neural Net with Clustering on Dimensionality Reduction for Diabetes Dataset

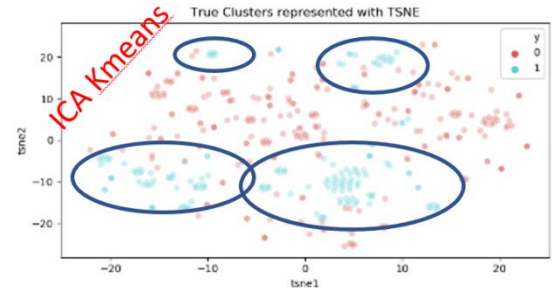
### Results /Analysis

Kmean + NN						GMM + KNN							
	precision	recall	f1-score	support	Confusion Matrix:		precision	recall	f1-score	support	Confusion Matrix:		
0	0.98	1	0.99	64	64	0	0	0.98	1	0.99	64	64	0
1	1	0.97	0.99	40	1	39	1	1	0.97	0.99	40	1	39
accuracy			0.99	104					0.99	104			
macro avg	0.99	0.99	0.99	104					0.99	104			
weighted	0.99	0.99	0.99	104					0.99	104			
PCA + Kmean + NN						PCA + GMM + NN							
	precision	recall	f1-score	support	Confusion Matrix:		precision	recall	f1-score	support	Confusion Matrix:		
0	0.98	0.98	0.98	64	63	1	0	0.98	0.95	0.97	64	63	1
1	0.97	0.97	0.97	40	1	39	1	0.93	0.97	0.95	40	1	39
accuracy			0.98	104					0.96	104			
macro avg	0.98	0.98	0.98	104					0.96	104			
weighted	0.98	0.98	0.98	104					0.96	104			
ICA + Kmean + NN						ICA + GMM + NN							
	precision	recall	f1-score	support	Confusion Matrix:		precision	recall	f1-score	support	Confusion Matrix:		
0	0.98	0.98	0.98	64	63	1	0	0.97	0.98	0.98	64	63	1
1	0.97	0.97	0.97	40	1	39	1	0.97	0.95	0.96	40	2	38
accuracy			0.98	104					0.97	104			
macro avg	0.98	0.98	0.98	104					0.97	104			
weighted	0.98	0.98	0.98	104					0.97	104			
RP + Kmean + NN						RP + GMM + NN							
	precision	recall	f1-score	support	Confusion Matrix:		precision	recall	f1-score	support	Confusion Matrix:		
0	0.98	1	0.99	64	64	0	0	0.98	0.95	0.97	64	61	3
1	1	0.97	0.99	40	64	0	1	0.93	0.97	0.95	40	1	39
accuracy	0.99	104			1	39			0.96	104			
macro avg	0.99	0.99	0.99	104					0.96	104			
weighted	0.99	0.99	0.99	104					0.96	104			
KCA + Kmean + NN						KCA + GMM + NN							
	precision	recall	f1-score	support	Confusion Matrix:		precision	recall	f1-score	support	Confusion Matrix:		
0	0.98	0.98	0.98	64	63	1	0	0.97	1	0.98	64	64	0
1	0.97	0.97	0.97	40	1	39	1	1	0.95	0.97	40	2	38
accuracy			0.98	104					0.98	104			
macro avg	0.98	0.98	0.98	104					0.98	104			
weighted	0.98	0.98	0.98	104					0.98	104			

The performances for all the combinations show that KMeans and EM are optimal without including any dimension reducing. Regular clustering performed even better than the regular neural nets. To understand the performances, we need to analyze the true clustering of these algorithm outputs. The true clustering holds the structure that is going into the neural nets. In addition, we will look at the type of errors that the clustering from reduced data produces.

From the first experiment, the resulting clusters for KMeans had errors defining the boundaries, but the true clusters have well-defined shapes. The same is said about the EM clusters. This gives us the hypothesis that if the resulting true clusters has well-defined shapes then the neural net will be able to correct the belief in training.

To show this this hypothesis, we look at the other algorithm true clusters and error types. The best example is KMeans on ICA, there are very little well-



	RP+NN	KPCA+NN	ICA+NN	PCA+NN	NN
Kmeans	0.3631	0.38	0.3939	0.3939	0.3949
EM	0.3919	0.3943	0.4009	0.381	0.3945
Normal	0.3949	0.3949	0.3561	0.392	0.39

defined features in the true clusters. This insinuate it will lead to less of the correct positives being selected and more false positives because the labels are so spread out. This is seen throughout all combinations of the dimension reducing algorithms, no matter if we are using KMeans or EM as the clustering method. This brings us to the conclusion that neural net can correct the fuzzy borders of the KMeans and EM outputs. However, if the data is more interweaved, the neural net performance decreases.

Time analysis (shown in the table snippet to the right) shows that ICA tends to take the longest and RP is, on average, the fastest algorithm when paired with NN and a clustering algorithm.

### References:

- [1] [https://github.com/ezerilli/Machine\\_Learning/tree/master/Unsupervised\\_Learning](https://github.com/ezerilli/Machine_Learning/tree/master/Unsupervised_Learning)
- [2] <https://archive.ics.uci.edu/ml/datasets/Early+stage+diabetes+risk+prediction+dataset>
- [3] [https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load\\_breast\\_cancer.html](https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html)
- [4] <https://machinelearningmastery.com/probabilistic-model-selection-measures/>
- [5] [https://scikit-learn.org/stable/auto\\_examples/cluster/plot\\_kmeans\\_silhouette\\_analysis.html](https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html)
- [6] <https://machinelearningmastery.com/how-to-reduce-model-variance/>
- [7] <https://ro-che.info/articles/2017-12-11-pca-explained-variance>
- [8] [http://labs.seas.wustl.edu/bme/raman/Lectures/Lecture14\\_ICA.pdf](http://labs.seas.wustl.edu/bme/raman/Lectures/Lecture14_ICA.pdf)
- [9] <https://towardsdatascience.com/visualising-high-dimensional-datasets-using-pca-and-t-sne-in-python-8ef87e7915b>
- [10] [https://en.wikipedia.org/wiki/Precision\\_and\\_recall](https://en.wikipedia.org/wiki/Precision_and_recall)