Unsupervised Learning

William Tidwell
Computer Science
Georgia Institute of Technology
Atlanta, GA, USA
trace.tidwell@gatech.edu

***Abstract*—The purpose of this assignment is to explore unsupervised learning techniques, specifically clustering and dimensionality reduction. For clustering, I considered k-means clustering and expectation maximization, and for dimensionality reduction, I considered principal components analysis, independent components analysis, randomized projections, and t-distributed stochastic neighbor embedding. For both datasets, I first conducted an exploratory data analysis using all the techniques listed above. Next, I clustered the data after having applied dimensionality reduction. Then, for one of the datasets, I analyzed the results of the neural network used in Assignments 1 and 2 after having applied the dimensionality reduction techniques to the data. Finally, I explored the effects of combining clustering and dimensionality reduction using the same neural network. A description of the problem and an analysis of the results are included for each part.**

***Keywords*—k-means clustering, expectation maximization, principal components analysis, independent components analysis, randomized projections, t-SNE**

I. Introduction

Up to this point we have dealt with labeled data, meaning we knew what the output of our learner should be. Many times, we are not so lucky, and we are given unstructured or unlabeled data. In these cases, we must first make sense of the data before can apply any supervised learning techniques. Two methods used for this are clustering and dimensionality reduction.

Clustering is used to find similarities in the data by grouping observations together based on some predefined distance metric. The bias of clustering is the choice of distance metric. Applying a traditional distance metric, such as Euclidean distance, on categorical data or text might yield no useful results, even if the some of the data is very closely related. k-Means clustering assigns each observation to one of k clusters. An observation is either in that cluster or not.

Expectation maximization uses distributions of Gaussians to assign a probability that each observation belongs to each of the k clusters, and each observation has a nonzero probability of belonging to each cluster. For observations that are far removed from the cluster centroid, the probability can approach zero. Expectation maximization can be useful when there are points that don't really seem to belong to any one cluster.

Dimensionality reduction can be used in a few different ways. As the name suggests, it can be used to try to reduce the number of dimensions, or features, of the data. Depending on the technique used, it can also be used to try to determine the importance of the features. In many ways these are the same, because if a feature is unimportant, it can be discarded, and, thus, the dimensionality has been reduced. The first technique discussed is Principal Components Analysis, or PCA. The goal of PCA is to find the components along the dimensions of maximum variance. In doing so, it is guaranteed to minimize error when reconstructing the original data from some set of reduced data.

The second technique is Independent Components Analysis, or ICA. ICA seeks to find the independent components of a dataset. Usually the dataset has been corrupted by noise, or the signals have been mixed together, or both. ICA works to separate the mixed signals back out into the original versions by making two key assumptions: that the sources are independent of one another, and that the source signals are represented by non-Gaussian distributions. As such, when transforming the data, ICA is essentially trying to maximize non-Gaussianity.

Randomized projection is the third technique used, and they are exactly as they sound. The data is randomly projected into some space with fewer features. If the data can be reconstructed from this projection well enough to still be useful, a reduction has been found.

The final technique is called t-Distributed Stochastic Neighbors Embedding (t-SNE). t-SNE is used to embed high-dimensional data in a low-dimensional space. t-SNE is a bit different than the other techniques, in that it is usually used to visualize high-dimensional data in a two- or three-dimensional space. In the case of labeled data used in classification problems, it can be used to look for trends or similarities in how the labeled data is grouped.

II. EEG Eye State

I originally chose this dataset because of work being done in human-computer interaction (HCI). People are working to use brain-computer interfaces (BCI) to detect brain waves and interpret them as speech. By mapping the signals coming from different parts of the brain to movements of body parts, we can effectively give the ability to speak to those who have lost the ability to move. This dataset is simply detecting if the eye is open or close, but it made me think of that research, and I can see how the two are related.

A. k-Means Clustering

k-Means clustering was the first algorithm applied to the EEG Eye State dataset. I used the Elbow Method to help determine how many clusters to use. The Elbow Method calculates the distortion of clustered data for a range of values of k. The distortions are then plotted, and the point at which the curve flattens out, or the “elbow”, is considered a good suggestion for how many clusters to use. Figure 1 below shows the results of the Elbow Method, and from this I chose k=4 to begin the analysis.

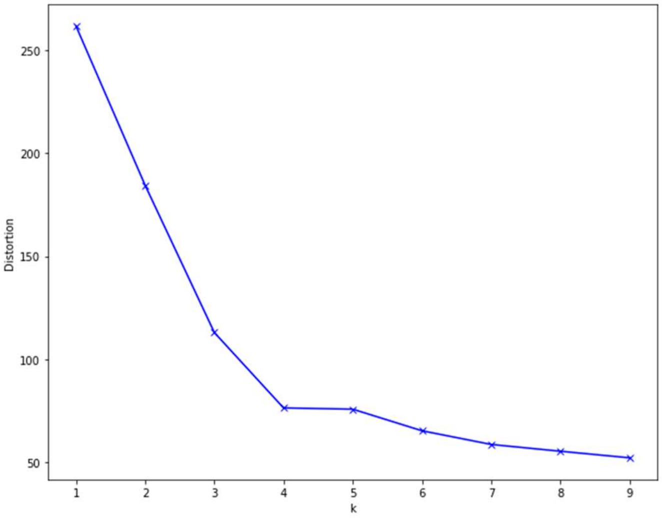

Fig. 1. EEG Eye State, Elbow Method, k-Means, All Data

After clustering the data, I wanted to see if there was any relationship between the cluster to which each observation was assigned and its class label. To my surprise, three of the clusters contained only one example each. A quick examination revealed that all three of these points are probably errors. Each of them contains attributes with values 2 orders of magnitude higher than the mean for that feature. Upon removing those three points, I ran the Elbow Method once more. The results can be seen in Figure 2 below, and this plot is much different than the first one. The distortion for k=1 and k=2 is almost the same before starting to trend downward. I decided to investigate clustering with k=2 and found, once more, what I believe to be an error. Cluster 2 contained only a single example, while Cluster 1 contained all the rest. While not as extreme as the others, it was still an outlier at best, so I decided to remove it.

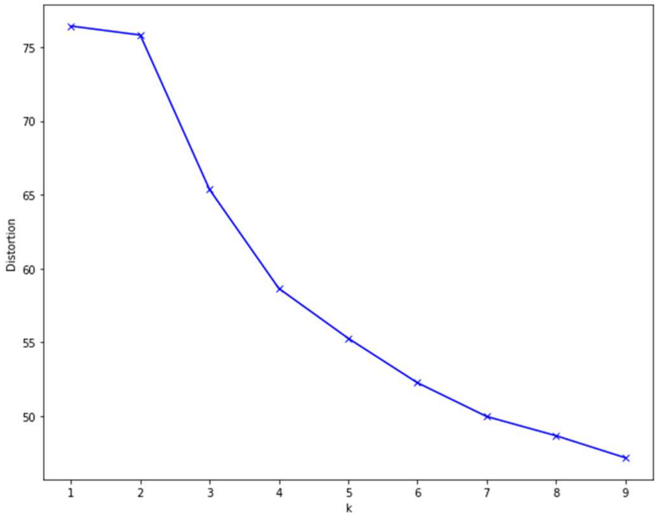

Fig. 2. EEG Eye State, Elbow Method, k-Means, -3 Errors

For the next iteration, I again ran the Elbow Method for up to 9 clusters. However, in this instance, there was not really a clear inflection point. The curve is relatively smooth all the way down to 9. To make sure I wasn't missing anything, I ran the Elbow Method again with up to 19 clusters and then 49 clusters. From Figure 3 below, we can see a bit of an inflection point at k=3 and at k=8. I clustered with each of these values, and I think there a few things worth noting.

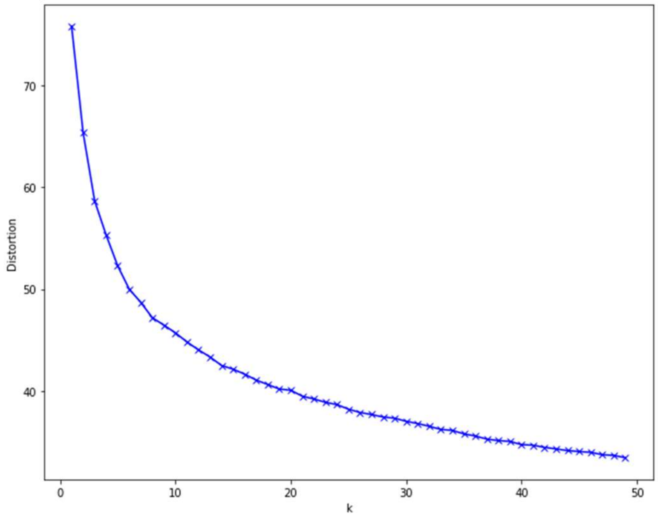

Fig. 3. EEG Eye State, Elbow Method, k-Means, No Errors/Outliers

Table I below shows the number of observations belonging to each cluster along with their respective classes. We can see that Cluster 1 contains 7,770 examples, or just over 50% of the dataset. Cluster 2 contains 1,099 examples, and Cluster 3 contains 6,107 examples. Unfortunately, each cluster contains a fairly even split of positive and negative labels, so this does not help to classify the data. If we were to assign the label of the majority class of each cluster to each element within that cluster, we could correctly classify nearly 56% of the data. Considering the data is slightly imbalanced with a 55/45 split, this really is not much better than just assigning the entire dataset the majority label.

TABLE I. EEG EYE STATE, K-MEANS, K=3

	Class 0	Class 1
Cluster 0	4620	3150
Cluster 1	621	478
Cluster 2	3013	3094

Next, I did the same for k=8 clusters. Table II below shows those results. We can see that Clusters 1, 3, and 5 dominate the others in terms of how many observations are assigned to them, collectively accounting for nearly 73% of the dataset. Assigning these clusters the majority label would result in an accuracy of 59.9%, which is better than before, but still not much better than taking the majority label.

TABLE II. EEG EYE STATE, K-MEANS, K=8

	Class 0	Class 1
Cluster 0	2007	1306
Cluster 1	52	443
Cluster 2	2319	1826
Cluster 3	403	108
Cluster 4	1874	1561
Cluster 5	198	210
Cluster 6	809	1126
Cluster 7	592	142

B. *Expectation Maximization*

Expectation Maximization (EM) was the second technique explored. Since each point has nonzero probability of belonging to every cluster, we cannot use the Elbow Method directly like before. Fortunately, there is something called the Bayesian Information Criterion (BIC). The BIC can be thought of as a way to quantify Occam's Razor, or trying to find the balance between model error and complexity. Indeed, the formula for BIC

$$BIC = \ln(n)\,k - 2\ln(\hat{L})$$

can be seen to be very closely related to that of the hypothesis with maximum a posteriori, or h_{MAP}

$$h_{MAP} = -\lg\big(P(D|h)\big) - \lg(P(h))$$

In the BIC formula, the term on the left represents model complexity, and the term on the right represents model error. For the h_{MAP} formula, the terms are reversed. As we increase the number of clusters, we expect BIC to decrease substantially at first and then to taper off. Using this, we can now apply the Elbow Method to EM. Figure 4 below shows the plot of BIC versus number of components. For each value of k, there are 4 ways of representing the covariance matrix, which is shown by the 4 vertical bars. We can see using the full covariance matrix results in a much lower BIC value across the board, so I decided to examine only those.

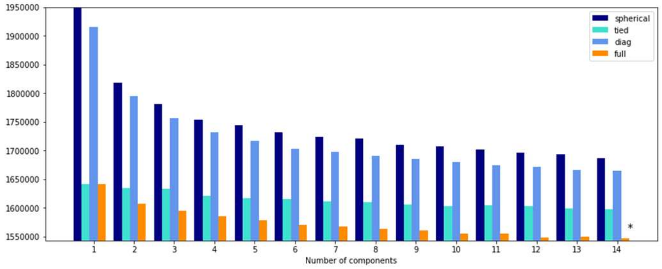

Fig. 4. EEG Eye State, BIC, EM

Figure 5 shows the plot of BIC using the full covariance matrix for up to 14 clusters. From this, we can see an inflection point at k=6 and one at k=12, where the curve really seems to flatten out. Since we examined k=3 and k=8 for k-Means clustering, and since k=6 is between the two, this seemed like an interesting size of k to examine further.

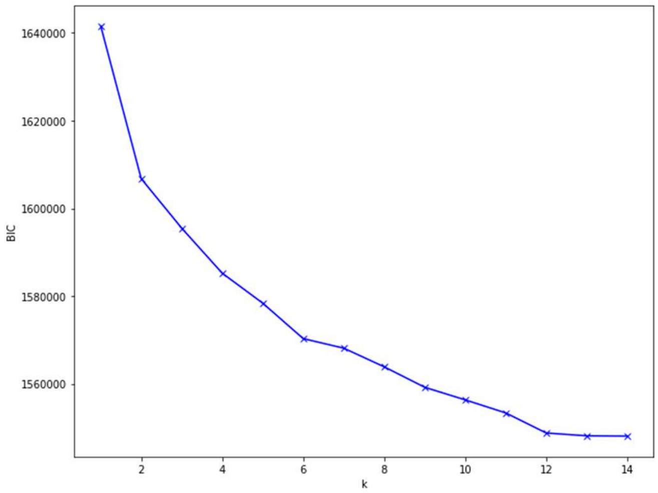

Fig. 5. EEG Eye State, Elbow Method, EM

Using the same approach as before, I split each cluster based on its class label. By taking the majority class label for each cluster, we can correctly classify 9,147 examples, or just over 61%. This is comparable to k=8 for k-Means above, but it still is not great.

TABLE III. EEG EYE STATE, EM, K=6

	Class 0	Class 1
Cluster 0	1639	635
Cluster 1	831	835
Cluster 2	4340	3265
Cluster 3	381	480
Cluster 4	1005	659
Cluster 5	58	848

C. *Principal Components Analysis*

As mentioned above, PCA seeks to maximize variance. Upon transforming the data, the first thing I noticed is that the first feature has an Explained Variance Ratio (EVR) of 63.0%, making it far and away the most important component. The next two components have EVRs of 13.8% and 12.0%, respectively. So, the first 3 components explain almost 89% of the variance, while the remaining 11 account for 11%. Depending on the size of the data and the desired accuracy of the model, we could potentially throw out a great deal of the data and still build a successful model. In fact, in keeping only features that have an EVR greater than 1%, we could get rid of 6 features, or nearly half the data, and still account for approximately 97% of the variance in the data! To test how well this works, I decided to transform the data and keep only 8 features. I was able to reconstruct the data with a sum of the root mean squared error value of 57.5. This comes down to a mean absolute percent error of between 0.02-0.11% for each of the features. Considering I dropped over 40% of the data, this is quite impressive. In the sections to follow, I will see how the neural network performs on the reduced dataset.

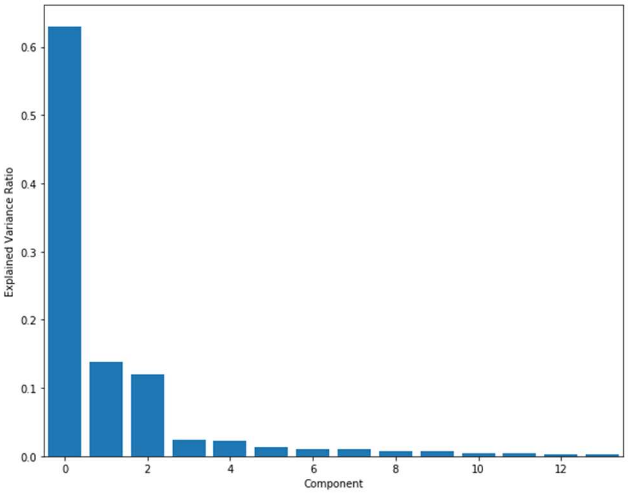

Fig. 6. EEG Eye State, PCA, EVR

D. *Independent Components Analysis*

From the description of ICA, we saw that it is attempting to maximize the non-Gaussianity of its components. A common way to measure the Gaussianity of a distribution is to measure its kurtosis, or the fourth standardized moment of a probability distribution, which measures the "tailedness" of the distribution. A Gaussian distribution has a kurtosis of 3. After transforming the data, I measured the kurtosis of the transformed features. I then subtracted 3 to measure the excess kurtosis, or each component's distance from a Gaussian. Components 0, 2, 5, and 11 are all very leptokurtic, meaning they have positive excess kurtosis, Components 3, 4, 6, 8, 10, and 13 are all moderately platykurtic, meaning they have negative excess kurtosis. Components 7 and 9 are the closest to Gaussians, so I decided to drop two components and see how well the data could be reconstructed. The reconstructed data had a total root mean squared error of 23.1, which is less than half of the error from PCA. This corresponds to mean absolute percent error of between 0.004-0.08% for each of the features. It is worth noting that here we only removed 2 features, whereas before we removed 6. However, with only 14 features total, even removing 2 features results in a 14% reduction in the amount of data.

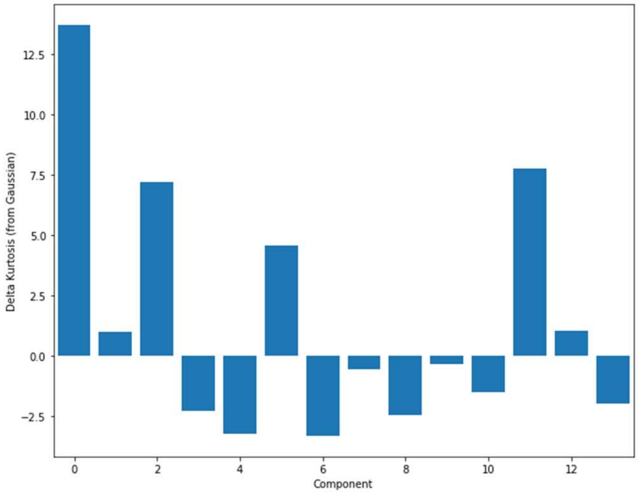

Fig. 7. EEG Eye State, ICA, Delta Kurtosis

E. *Randomized Projection*

Randomized projections did not prove fruitful for this dataset. I experimented with a Gaussian Random Projection and a Sparse Random Projection, neither of which had very promising results. Multiple projections with both methods resulted in total mean squared errors of between 35,000 and 70,000+, with MAPE values as little as 3.5% and as large as 400+%. For this dataset, at least, I would not use randomized projections.

F. *t-Distributed Stochastic Neighbor Embedding*

t-SNE was used to reduce the dimensionality of the dataset from 14 down to 2 while still maintaining the relationships between the observations as much as possible. The figure below is the t-SNE plot of the EEG Eye Movement data, with the different colors representing the different labels. While there are a few parts where the data very clearly belongs to one class or the other, most of it mixed together in the center of the plot. Though I am not sure that I could do much with this on its own, combined with the other methods used above it helps to confirm the earlier findings: that this dataset is not easily separable into its two classes.

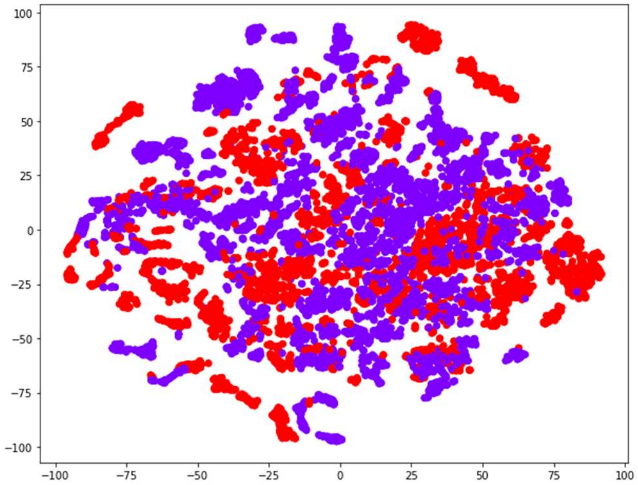

Fig. 8. EEG Eye State, t-SNE

G. *k-Means after PCA*

The next part of the analysis involves running the clustering algorithms on the data after having applied dimensionality reduction techniques. Much like before, the curve is smooth, with no defining elbow. There does appear to be an inflection point at k=8, and we analyzed k=8 for k-Means before applying dimensionality reduction, so to keep things consistent I decided to further explore that value of k.

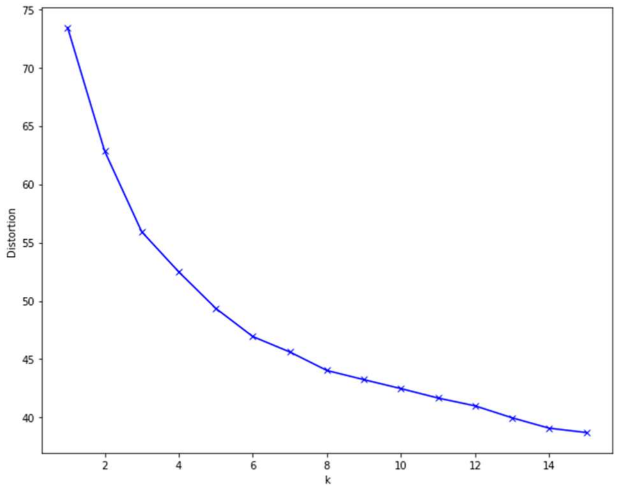

Fig. 9. EEG Eye State, Elbow Method, k-Means on PCA

The breakdown of the distribution of classes amongst the different clusters is very much the same as it was before dimensionality reduction. Three of the clusters contain over 70% of the data, and the accuracy rate achieved by assigning the majority label is 59.9%. In fact, from Section A above, we would classify 8,978 observations correctly, and here we would classify 8,975 observations correctly.

H. k-Means after ICA

We will now apply k-Means clustering the data that has had ICA performed. In this case, we transformed the data by reducing it from 14 components to 12. Figure 10 shows the Elbow Method plot. This plot has the smoothest curve seen thus far. It has become increasingly difficult to determine a cutoff point for a good number of clusters. I think this just goes to show how difficult this dataset is to cluster properly. Just to compare to the previous iterations of k-Means, I calculated the classification accuracy of k=8 clusters. Much like before, it could correctly classify around 60% of the observations. What's interesting in this case, though, is that split between label 0 and label 1 changed, and this clustering would classify far more observations from label 1 correctly, at the expense of correctly classifying observations from label 0, of course.

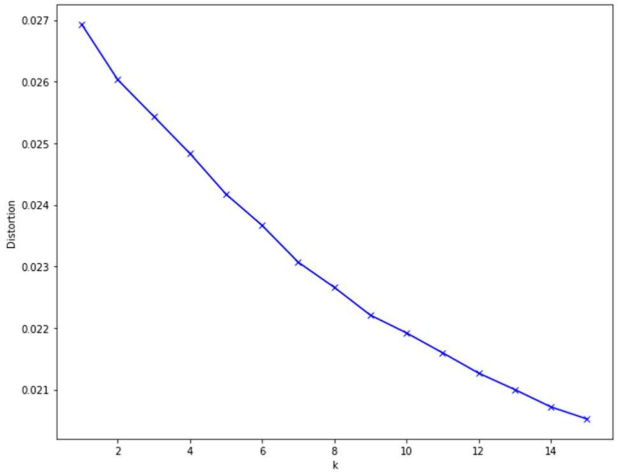

Fig. 10. EEG Eye State, Elbow Method, k-Means on ICA

I. EM after PCA

Using the same approach as before, I first ran the BIC analysis to determine which covariance matrix to use. This is shown in Figure 11 below. Once again, the full covariance matrix is much better than the others for k > 1.

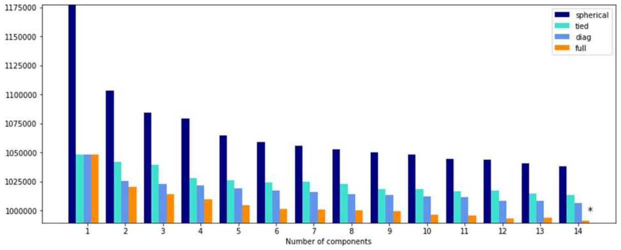

Fig. 11. EEG Eye State, BIC, EM on PCA

Using the full covariance matrix, I plotted the BIC values of each mixture model for k=1 to k=14 components. Once again, there is clearly an inflection point at k=6. In this case, the results are worse than before performing PCA, with an accuracy of 58.2% versus 61.1% before performing PCA.

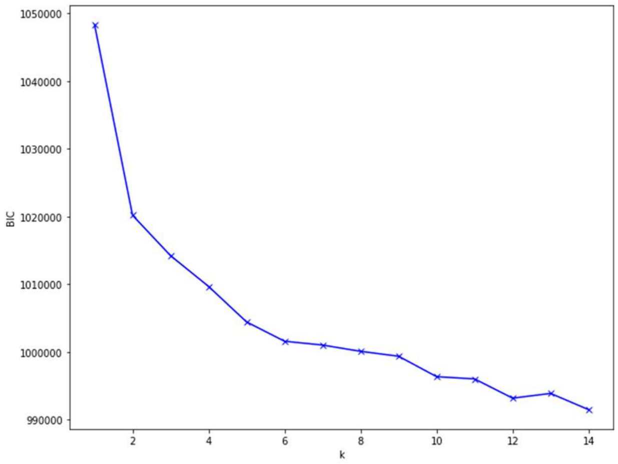

Fig. 12. EEG Eye State, Elbow Method, EM on PCA

J. EM after ICA

The BIC plot for EM following ICA is a little different than those above, and the BIC values become quite negative. Since the data was transformed and took on new values, many of which are now negative, I believe this has more to do with the transformed data than it does with the model itself now being much better than before.

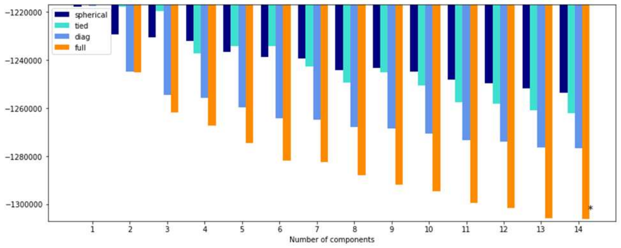

Fig. 13. EEG Eye State, BIC, EM on ICA

Finally, a plot of the BIC values using the full covariance matrix is plotted below for a range of values for k. As we may have expected at this point, the Elbow Method seems to indicate that k=6 is a good number of components to use. As with the other clustering methods and values of k, we can correctly classify about 58.7% of the data, and three of the clusters contain the majority of the data.

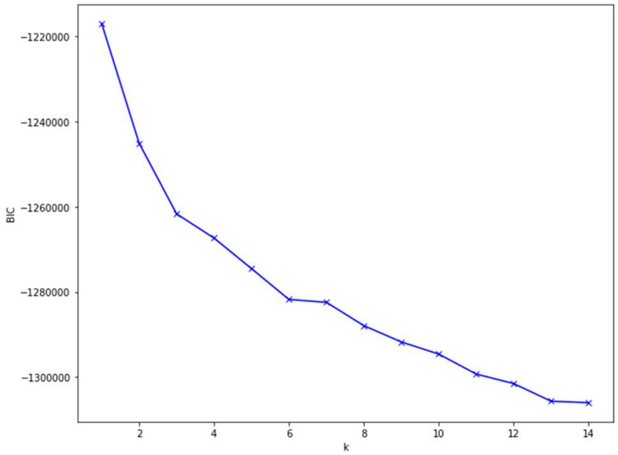

Fig. 14. EEG Eye State, Elbow Method, EM on ICA

K. *Neural Network with Dimensionality Reduction*

Next, I examined the effects of dimensionality reduction on the learners. First, I trained the neural network used in Assignments 1 and 2 on data after applying PCA. From before, we saw that we could drop over 40% of the data (6 of 14 features) and still explain 97% of the variance. Using PCA and keeping only those 8 components resulted in greater accuracy on the neural network than before by a considerable amount. A comparison of the metrics is outlined in Table IV below. Accuracy improved by nearly 5%, precision improved by nearly 7%, recall improved by 4%, and the F1 Score improved by nearly 5%. Throwing out nearly half the data and seeing almost 5+% improvements on all metrics truly is a testament to how bad the Curse of Dimensionality really is. For further comparison, I checked to see which of the algorithms performed the best on this dataset from Assignment 1. KNN has an accuracy 0.9719, with the other metrics all around 0.97. However, KNN on 100% of the data is much, much slower than a neural network on 58%.

Next, I ran the neural network after performing ICA. For ICA, I opted to go from 14 features to 12, a reduction of about 14%. The results here are even more impressive than with PCA, with all the metrics reaching 95% and the accuracy going over 96%. This is an improvement of 7-10% for all metrics, and the AUC is 0.99, which matches that of KNN from Assignment 1. The network is now rivaling the performance of the best learner on only 86% of the data.

Finally, I decided to run the neural network on the features derived from t-SNE. t-SNE is not always used train learners after being used as a dimensionality reduction technique, but if it can capture the relationships in the underlying data well enough, it can certainly be applied. In this case, training the neural network on the t-SNE features resulted in scores that essentially matched those of the network trained on all the data. It's worth stating explicitly here that this was achieved using only two features! With just a bit of preprocessing, I can achieve the same results using only two features as I did using the entire dataset before.

TABLE IV. EEG EYE STATE, NN WITH DIMENSIONALITY REDUCTION

	No DR	PCA	ICA	t-SNE
Accuracy	0.8870	0.9346	0.9617	0.8892
Precision	0.8591	0.9278	0.9598	0.9099
Recall	0.8839	0.9211	0.9519	0.8270
F1 Score	0.8713	0.9244	0.9558	0.8665

L. *Neural Network with Dimensionality Reduction and Clustering*

After establishing that dimensionality reduction can improve the performance of supervised learning techniques, and that clustering can help to explore our data, we wanted to examine whether combining the two can further improve the predictive capabilities of the learners. Unfortunately, applying the neural network to just the clusters derived using k-Means and EM on both of PCA and ICA all resulted in an accuracy hovering around 60% across the board. 60% accuracy is what we have seen throughout the clustering exercises and can be achieved by simply assigning each observation in the cluster the majority class label. I suspect that, with only one feature to go on and how weak the clustering has been, the learner did just that. It's worth mentioning here that, in addition to having the highest accuracy, training on the data clustered using k-Means after applying ICA offers the balanced results, as evidenced by the precision, recall, and F1-score. This means it's not achieving its results by assigning everything to one class. Something in the combination of techniques is allowing it to classify observations from classes pretty evenly.

TABLE V. EEG EYE STATE, NN, DIM. RED., CLUSTERING

	k-M/PCA	EM/PCA	k-M/ICA	EM/ICA
Accuracy	0.5986	0.5826	0.6217	0.5919
Precision	0.5895	0.5675	0.5699	0.6948
Recall	0.2528	0.1679	0.5301	0.1095
F1 Score	0.3539	0.2591	0.5493	0.1892

To further test the combination of clustering and dimensionality reduction, I decided to add the cluster labels to the reduced datasets after applying each of PCA and ICA. In both cases, the accuracy went down slightly. Adding the k-Means cluster labels to the PCA reduced data resulted in a 93.1% accuracy, down from 93.5%. Adding k-Means cluster labels to the ICA data resulted in a 95.4% accuracy, down from 96.2%. As one last test, I decided to first cluster the data, add the cluster labels as features, and then run PCA to see if there are any performance gains to be had. However, once again, the clustered data did little to nothing. The best principal components remained the same, so the cluster labels would not be included in the reduced data.

III. Bank Marketing

I chose the bank marketing dataset because it had the same number of features as the EEG Eye State dataset but had roughly 3 times the data. I was interested in seeing how the Curse of Dimensionality came into play. What I did not consider was the number categorical features or how imbalanced the dataset is, with roughly 89% of the observations belonging to one class. Regardless, since my other dataset did not contain those features, I decided to stick with it to see what kind of differences I might find.

A. *k-Means Clustering*

k-Means clustering was once again the first algorithm applied. The Elbow Method was employed to determine the number of clusters to use. There point at k=6 is very distinct, and I think the argument for k=8 could also be made, but I chose k=6. This is shown in Figure 15 below.

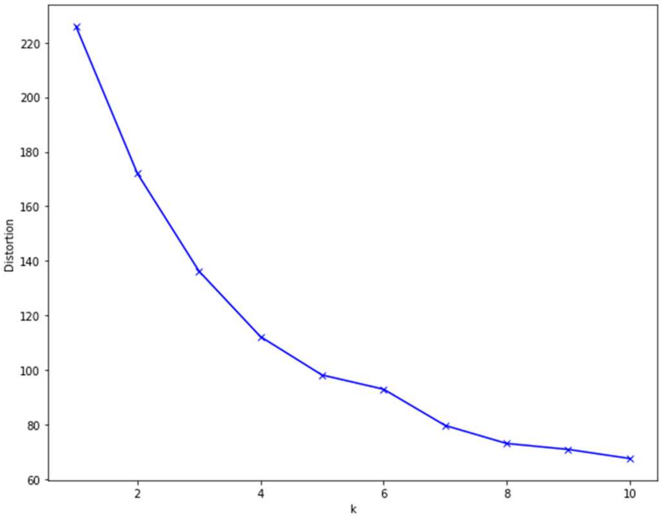

Fig. 15. Bank Marketing, Elbow Method, k-Means

Like with the EEG Eye State data, I examined whether there was relationship between the cluster labels and the class labels. The first thing I noticed is that cluster 3 correctly classifies 99.8% of the observations, and the cluster itself accounts for over 54% of all observations. Upon digging into the data, I found that the most likely cause for this split is the "duration" feature, which measures, in seconds, the duration of the communication the bank had with the potential customer. The mean duration time for Cluster 3 was 108 seconds, while for all other groups it was 431 seconds. Considering the purpose of the exercise is predict if the client will subscribe to a term deposit, it makes sense that someone who is less willing to spend time to hear about a product is also less likely to buy it.

TABLE VI. Bank Marketing, k-Means, k=6

	Class 0	Class 1
Cluster 0	10553	1116
Cluster 1	544	958
Cluster 2	3151	1199
Cluster 3	21954	438
Cluster 4	628	798
Cluster 5	78	131

B. *Expectation Maximization*

The second clustering algorithm was Expectation Maximization. I calculated the BIC using up to 14 Gaussians to cluster the data. Figure 16 below shows the BIC values using the different types of covariance matrices. Once again, using the full covariance matrix for each Gaussian results in the lowest BIC. What is new about this plot is that for all k > 1, the BIC using the full covariance matrix is negative, and for k > 9, only the spherical covariance matrix is positive, and it is much, much larger than with the other three.

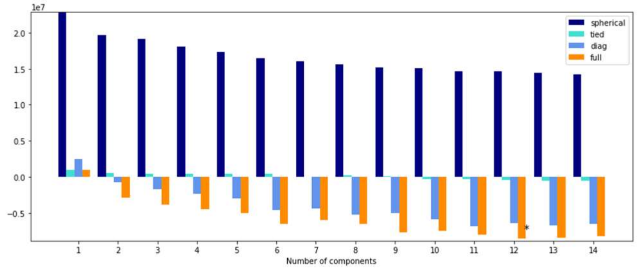

Fig. 16. Bank Marketing, Eye State, BIC, EM

From the Elbow Plot made from the BIC values obtained using the full covariance matrix, we again see that k=6 seems to be a good fit. Unlike before, rather than k=8 as another potential option, here k=9 looks like a good possibility. However, the increase at k=7 and again at k=10 is very peculiar. This seems to indicate that any reduction in error (if any) by moving from 6 components to 7 is offset by additional model complexity.

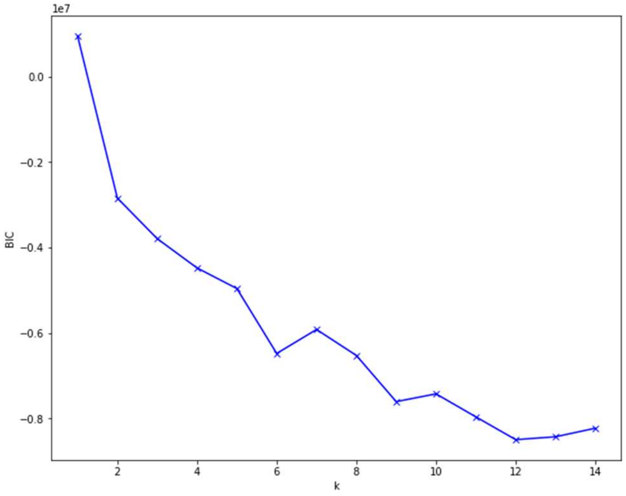

Fig. 17. Bank Marketing, Elbow Method, EM

In examining the class labels of the clusters, the results are very similar to the ones obtained using k-Means above in that a single cluster accounts for a large share of the data and correctly classifies a high percentage of observations. In this case, Cluster 4 contains almost 63% of the data and correctly classifies over 95% of the observations. Interestingly, though, the features that seem to explain this differ from the one(s) found using k-Means. This time, the number of times the customer had been contacted during previous marketing campaigns and unemployment rate seemed to be the biggest predictors. More specifically, the members of Cluster 4 were far less likely to have been previously contacted and were incredibly less likely to be unemployed. In fact, only one person belonging to Cluster 4 was unemployed, compared to 1,013 for the remaining 5 clusters.

TABLE VII. BANK MARKETING, EM, K=6

	Class 0	Class 1
Cluster 0	7039	963
Cluster 1	1462	802
Cluster 2	545	963
Cluster 3	2711	643
Cluster 4	24660	1190
Cluster 5	131	79

C. *Principal Components Analysis*

Now we move to PCA on the Bank Marketing Data. Figure 18 shows the Explained Variance Ratio of each component. What we quickly see is that only 4 components are visible on the chart. Actually viewing the numbers, it becomes clear that the first two components explain 95% of the variance, and the first three explain 99.9% of the variance. The dataset has 53 features, and PCA is telling me I can throw away 50 of them! After performing dimensionality reduction down to three features and reconstructing the data, I have total mean squared error of 31.49. Due to the sparsity of the data in some areas from the categorical features, I was unable to calculate percent error because of divide by zero errors.

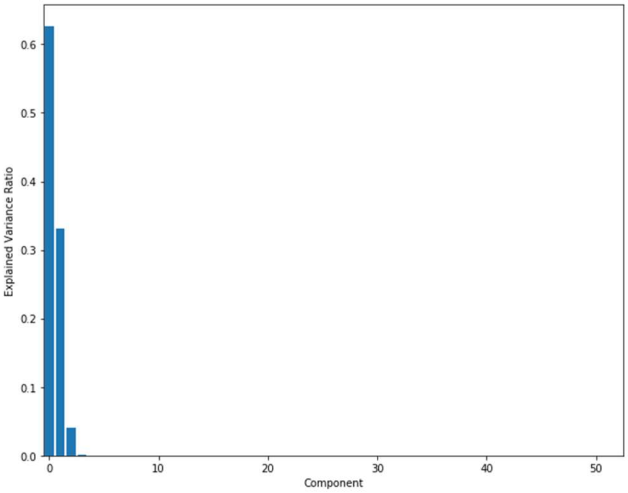

Fig. 18. Bank Marketing, PCA, EVR

D. *Independent Components Analysis*

Next, I ran ICA on the Bank Marketing data. Here again two of the components leap off the page as being far more important than the others, with a few more perhaps worth considering. In fact, the most important component has a kurtosis nearly an order of magnitude larger than that of the second most important component, which is itself an order of magnitude larger than the next best. After reducing the dataset down to 5 features, I was able to reconstruct the original dataset with a total mean squared error of 16.28. This is compared to 31.49 using PCA above, but we used two more features. Both results are pretty impressive.

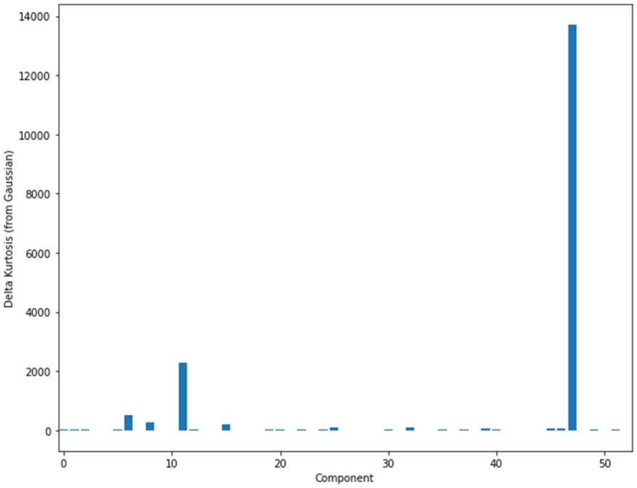

Fig. 19. Bank Marketing, ICA, Delta Kurtosis

E. *Randomized Projection*

For the randomized projections I again used both a Gaussian Random Projection and a Sparse Random Projection. Based on what I've learned above, it doesn't seem as though most of the features are needed. Since the random projections are, well, random, I decided to keep a few more features in hopes that it helped the performance. I experimented with 10, 15, and 20 features for both methods. With the Gaussian Random, my total mean squared error seemed to stay between 40,000-60,000 for all three number of components, occasionally dipping into the 35,000-40,000 range. However, the Sparse Random projection fared a little better than the Random Gaussian (though still did terribly compared to PCA and ICA above). Using 10 components I had multiple projections with error below 10,000. Unfortunately, many of them had errors in the 60,000-80,000 range, too. My guess is that, with all the categorical variables, the data itself is sparse in areas, and a few of the sparse projections matched up well with the data. As before, especially with how well PCA and ICA performed, I would stay away from randomized projections.

F. *t-Distributed Stochastic Neighbor Embedding*

The final dimensionality reduction technique used on the Bank Marketing data. The plot can be seen in Figure 20. Unlike with the EEG Eye State dataset, there are very clear pockets that belong to only one class. Because the dataset is so imbalanced, most of the groupings clearly belong to Class 0, but there are some distinct pockets where Class 1 are prevalent. In a way, this lines up very nicely with the clustering algorithms from above. When we examined the class labels of the clusters, we saw that two of the clusters were dominated by Class 0 and one was heavily skewed towards Class 0. But out of the other three, two were pretty evenly mixed, and one had Class 1 as the majority label. For example, when using EM, the area in the bottom left of the plot could correspond to Cluster 2, while the areas to the left and top might correspond to Clusters 1 and 5.

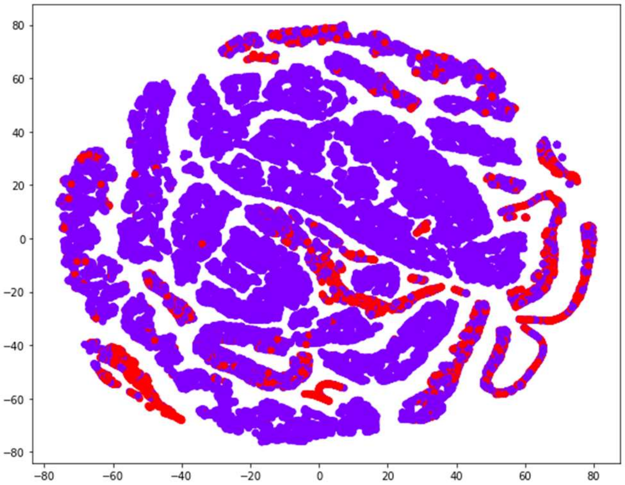

Fig. 20. Bank Marketing, t-SNE

G. k-Means after PCA

Now we will perform k-Means clustering on the dataset after applying PCA. We have reduced the dataset from 53 features down to only 3. The curve looks very much like the one from above, and while I think k=6 is again a good number of clusters, this time the data seems to flatten out a bit more at k=8, so that's what we'll look at.

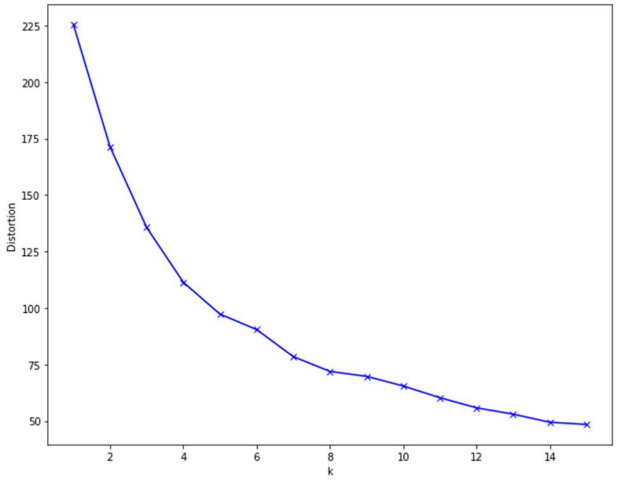

Fig. 21. Bank Marketing, Elbow Method, k-Means on PCA

The distribution of the class labels over the clusters is still very much the same as before, even with the two additional clusters. The smallest cluster has only 102 observations, so it could be argued that k=7 would have been more appropriate, However, that small cluster is made up primarily of observations from Class 1, the minority class. So maybe it is providing some information after all. In total, the clusters could classify about 90% of the data correctly, which may sound good, but with the imbalance it is still only slightly better than assigning everything to Class 0.

H. k-Means after ICA

Now we will cluster the data reduced using ICA with k-Means. The Elbow Method is shown in Figure 22 below. Due to the flattening of the curve between k=9 and k=10, I would say the elbow occurs at 9 components. When examining the class labels of the clusters, we again see that Class 1 is almost not represented at all. In only one of the 9 clusters is Class 1 the majority label, and it only contains 966 observations from that class.

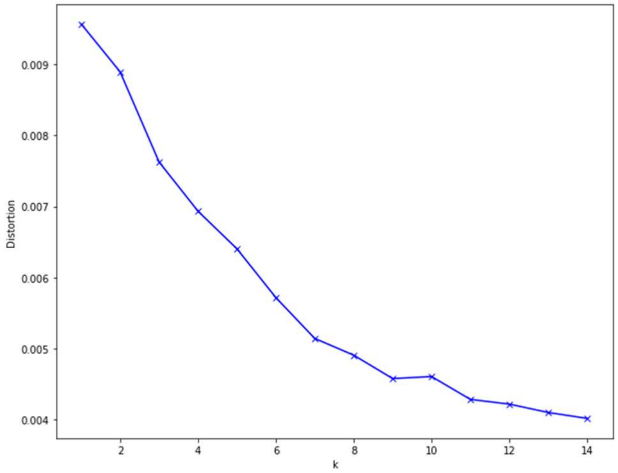

Fig. 22. Bank Marketing, Elbow Method, k-Means on ICA

I. EM after PCA

The first step in determining the correct number of components in EM is to plot the BIC values. As has been the case the whole time, using the full covariance matrix yields the best results. Even with the small picture, we can see that the BIC value drops off with k > 1 and remains steady after k > 4.

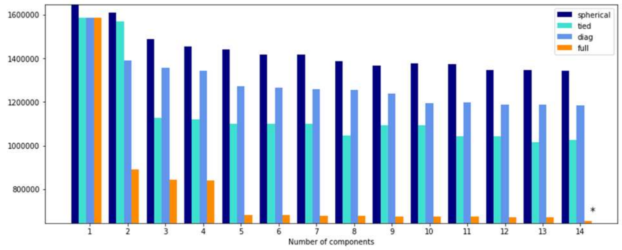

Fig. 23. Bank Marketing, BIC, EM on PCA

The Elbow Method was plotted using the BIC values and the full covariance matrix just to make sure we are not missing anything. The plot in Figure 24 lines up nicely with what we saw in the full plot above. This time we choose 5 components instead of 6, as was the case without PCA. The interesting thing about these clusters is that, while there is the one that absolutely dominates for Class 0 (99.3%), 3 of the 4 remaining clusters are relatively balanced. However, the reduction in clusters may have also resulted in a lower accuracy, because this grouping would only correctly classify 20% of Class 1.

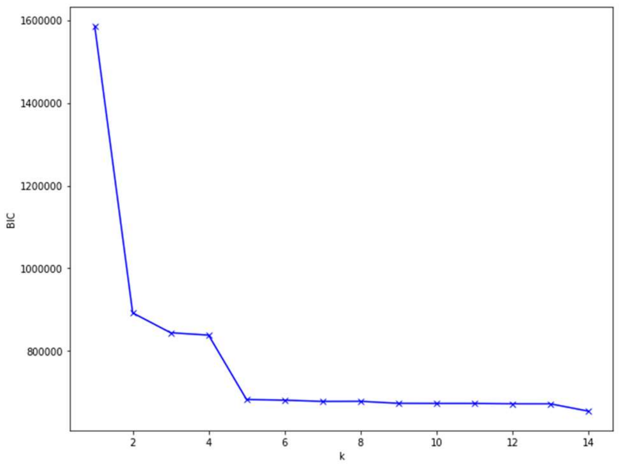

Fig. 24. Bank Marketing, Elbow Method, EM on PCA

J. *EM after ICA*

Just as with the EEG Eye State data, the BIC values for EM clustering after applying ICA are all negative. Figure 25 below shows this plot. From the figure we can see that the BIC using full covariance increases slightly at k=2 before decreasing, then increases again at k=6, before leveling off.

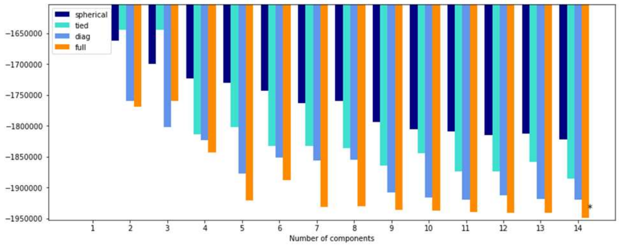

Fig. 25. Bank Marketing, BIC, EM on ICA

Figure 26 below does a better job of showing the fluctuations in the BIC values. This trend also resembles the plot from the EEG Eye State data from above. With the peak at k=6, and especially because the values level off starting at k=7, I would set the number of components to be 5 for this case. The overall accuracy has once again dropped.

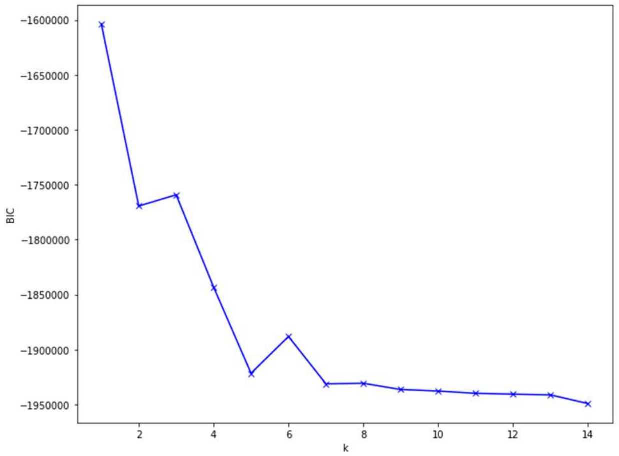

Fig. 26. Bank Marketing, Elbow Method, EM on ICA

IV. Conclusion

The first thing I noticed, for these two datasets at least, is that dimensionality reduction can be incredibly powerful. For the EEG Eye State dataset, PCA reduced dimensionality by 43%, and ICA reduced dimensionality 14%, and both led to increases in the performance of the neural network. There were very clearly features that were unnecessary that affected the ability of the neural network to learn. For the Bank Marketing dataset, the results of dimensionality reduction were even more extreme, indicating that over 90% of the data was irrelevant. And while the performance of the neural network was not improved, achieving the same results on only 5% of the data is remarkable. I would be interested to experiment with adding some features back in to see if the accuracy of the model can be improved.

I also learned that t-SNE can be beneficial for both visualization and feature extraction. It's a quick way to get an overview of how the data is related in a two-dimensional space that can be comprehended by humans. For these two datasets, the results aligned very closely with those obtained from the clustering algorithms, so it's a great way to start any analysis that involves clustering. The features extracted using t-SNE also proved to be good ones. For the EEG Eye State dataset, I achieved the same results using the two t-SNE features as I did using the full dataset. Though not discussed above, I was curious if this would work on the Bank Marketing dataset, as well. The results were not as promising, but even the possibility of training on only 2 features instead of 52 is worth exploring.

The second major point worth discussing is that, while dimensionality reduction proved very fruitful, clustering did not seem to provide much insight. I would not say it was completely useless, but I think it depends very much on the problem trying to be solved. In the case of having labeled data, including clustering information made little difference on the performance of the learner. In some cases, performance degraded. However, there were some benefits. For the EEG Eye State dataset, it allowed me to identify errors and/or outliers in the data that had gone undetected before. For the Bank Marketing dataset, it helped me to identify 2 distinct clusters that correctly classified over 50% of the data with 99% and 95% accuracy, respectively, and their respective potential causes, which seemed to differ based on the clustering algorithm. These types of insights could prove very useful moving forward, because it could tell the marketing team where they need to focus their efforts.