Name: Het Piyush Sheth **GTID:** 903393644

CS 7641-HW3

Introduction

This assignment deals with Clustering and Dimensionality Reduction problems. This report is divided into six sub-sections. The first sub-section gives a brief introduction of the datasets used. The second sub-section explores the two clustering algorithms (K-means and EM). The third sub-section explores the performance of four dimensionality reduction algorithms on the two datasets (PCA, ICA, RP, RF). The fourth sub-section evaluates the clustering algorithms on the data obtained after dimensionality reduction. The fifth sub-section evaluates the performance of the previously used neural network (from assignment 1) after applying the dimensionality reduction algorithms on the dataset. The last sub-section evaluates the performance of the same neural network after passing the data through the clustering algorithms. I have executed the above algorithms using the Scikit-Learn library with Python.

Dataset description (Section 1)

The two datasets that I used for this analysis are Breast Cancer Dataset and Digits Dataset. Before I describe the datasets itself, I want to talk about why these datasets. Earlier, I used the Adult and Breast Cancer dataset. Now, both of these were binary classification problems and I wanted to explore the possibility of multiple classes in my analysis (thus the digits dataset). And obviously, I wanted to keep one binary class dataset in my analysis too (to see what the algorithms do on the more obvious problem of the two.), thus the Cancer dataset.

The Cancer dataset has 700 instances (approx.) with 10 numeric features and two output classes (malignant and benign). The interesting part about this problem is that the clusters are not completely disjoint, thus giving out an initial poor performance. I believe that this poor performance leaves a lot of scope for improvement and will help enrich my analysis and provide some interesting results. The Digits dataset has 1797 instances with 64 features and 10 classes (digits from 0 to 9). This dataset is derived from an image dataset of the handwritten digits. An image is split into 8x8 grids and each cell within that grid is considered as a feature (thus giving 8x8 = 64 features). In addition, there are approximately 180 instances of each digit adding up to a total of 1797 instances. This means that the dataset is balanced and produces good results on the initial algorithm. Thus the stark contrast between the two datasets will be interesting to analyze.

Clustering (Section 2)

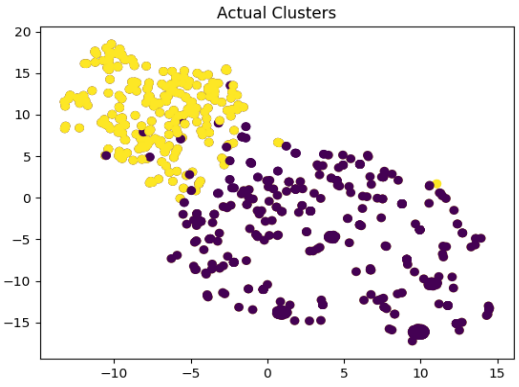
Clustering refers to forming groups from the given data points. The points that belong to the same cluster are supposed to be similar to each other than the ones far apart. It comes under unsupervised form of learning. The two clustering algorithms that I analyze in this report are k-means and Expectation Maximization.

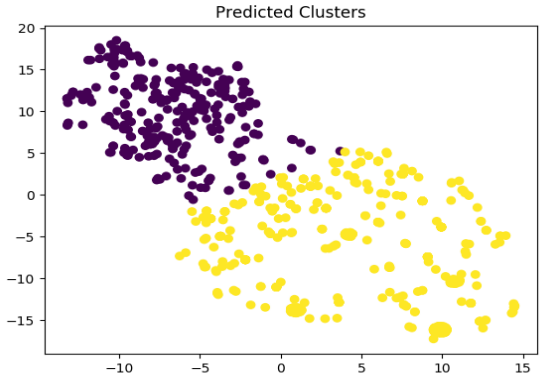
K-means

K-means algorithm segregates the data into k different clusters. Initially, it starts with random assignment of clusters but over time it iterates by taking centroids of clusters and re-computing the distances between data points and centroids of each cluster to form new clusters. Ideally, this continues iterating until it reaches a local optimum. I tried this algorithm on the two datasets and used several metrics to measure the performance on each dataset.

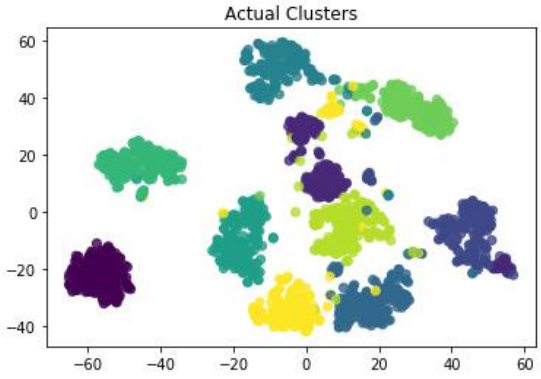
For both the datasets, I used Euclidian distance as a measure of the distance used by the K-Means algorithm. This is because the feature values for both the datasets are numerical values itself. I would consider using a different distance measure if the values were categorical or ordinal in nature. Although, this is an unsupervised learning problem, I used the dataset with the labels as it would help me analyze results. First, I performed K-means on the

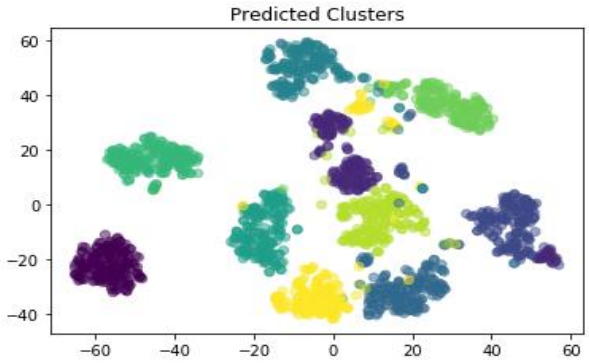
breast cancer dataset and plotted out the actual vs predicted clusters. Overall, the performance of the algorithm is good but it fails to correctly segregate a few of the data points that lie in the vicinity of the second cluster. This is because K-means is primarily a distance based clustering algorithm. Thus if data points are sparsely placed as for the bottom left cluster in the image, then K-means might not give the best results.





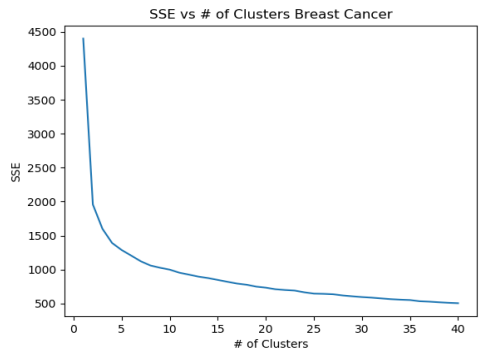
A similar observation is made for the digits dataset. That is, the clusters that are less spread out are better obtained by the K-means algorithm (sea-green, purple) as opposed to the ones that are close together (green, light-blue and purple). Conclusively, we can observe 10 distinct clusters in the predicted graph. Thus, indicating that overall, the algorithm performs pretty well.

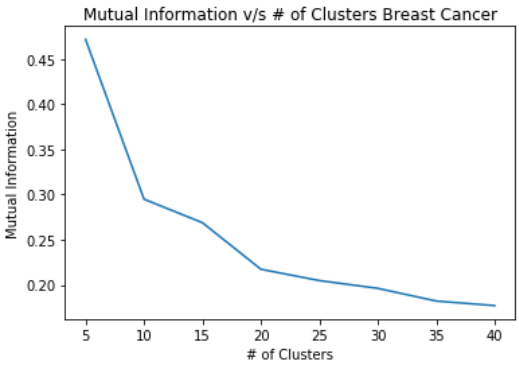




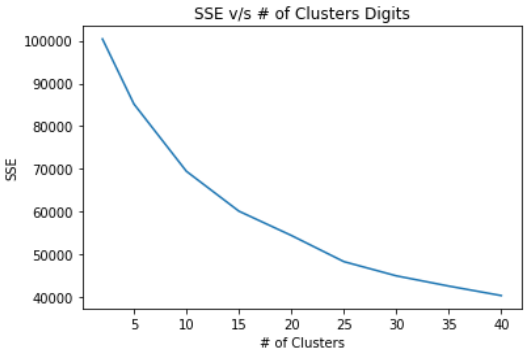
In order to evaluate the performance of the K-means algorithm I analyzed the Sum of Squared Errors (SSE) and Adjusted Mutual Index. I chose SSE as a metric of evaluation because K-means is inherently a distance based clustering algorithm. And thus SSE is the best way to measure the sparsity of the data points with respect to the clusters. In addition, mutual information represents the closeness between two clusters. The greater the value of MI, the closer are the clusters and vice versa. Thus, an ideal algorithm should produce clusters with low mutual information.

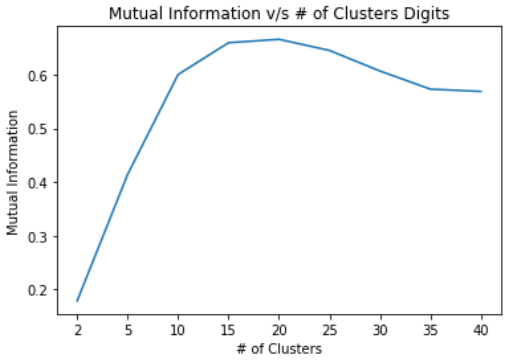
Below I have plotted the SSE and adjusted mutual information for the cancer dataset. We can see that for both the graph, there is a steep decrease initially and slow decrease after that. Now, eventually if we set the value of the number of cluster equal to that of the number of data points, then the error will be zero. But this only means that we are considering every data point as a cluster itself and thus, we are clearly overfitting the dataset. So our ideal number of clusters would be the point where the steep decrement stops and slow decrement starts as we treat the slow decrement as a case of overfitting. This method is also known as the elbow curve method. Clearly, from the graphs below, we can see that the ideal number of clusters based on SSE is somewhere between 0 and 5 (which is correct as the actual number of clusters should be 2). At the same time, mutual information keeps on decreasing steeply for a while greater than the SSE, this is because the clusters are not inherently separable (i.e. they have overlap data points). Thus, even though the best number of clusters that can be formed is true, the mutual information does not decrease for a while.





As for the digits dataset we can see that the SSE curve follows a similar trend with a sharp point at number of clusters = 10 (i.e., the point where the steeply decreasing curve becomes a slowly decreasing curve). But the mutual information graph is completely opposite to what was observed in a cancer dataset. This is because; the digits dataset has multiple clusters that are spaced closely to each other. Thus, the mutual information increases overtime indicating close proximity of multiple clusters.



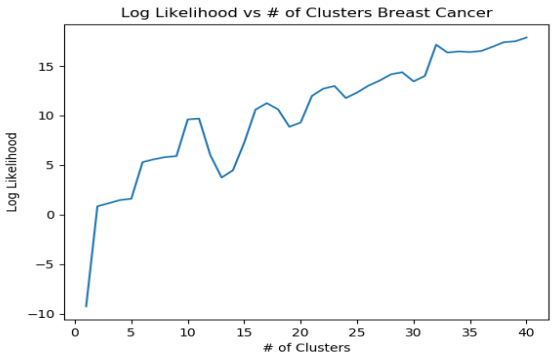


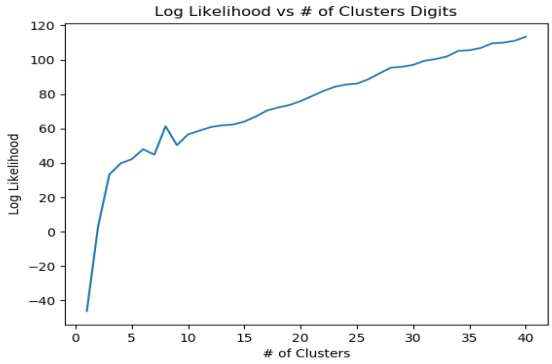
Expectation Maximization

This algorithm performs the clustering of data points in two steps. The first step is expectation, i.e, finding the probability (likelihood) that a point i belongs to a cluster j, and the second step is maximization, i.e, checking that a particular data point came from a given cluster. These two steps in iteration are performed to find stronger clusters till we reach a local maximum. Alternatively we can say that EM is performed to find the maximum likelihood estimates for parameters in probabilistic models. I performed EM on the above two datasets to generate clusters and measured the performance using log likelihood as a metric.

We can see below that for both the datasets, the log likelihood increases as we increase the number of clusters and it reaches a maximum for number of clusters = no. of data points. This is a clear case of overfitting for the reason explained above. Thus, for the ideal number of clusters, we should stop at the point where the steep increase gets converted to a slow increase. For the cancer dataset we can see that it is very soon for number of clusters < 5. Whereas there is a sudden peak and drop at number of clusters = 10 for the digits dataset (which in fact is the correct number of clusters.)

One other thing that I can infer from these graphs is that the curve is unstable for the cancer data once it crosses the ideal number of clusters and it is stable for the digits data. This is because there are more number of wrong categorizations in the cancer dataset and thus the likelihood fluctuates because of them (This cannot ideally be rectified as the points overlap, which could be a strong indication of an outlier too!).

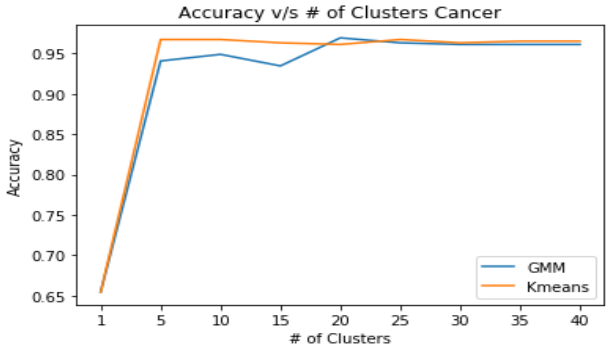


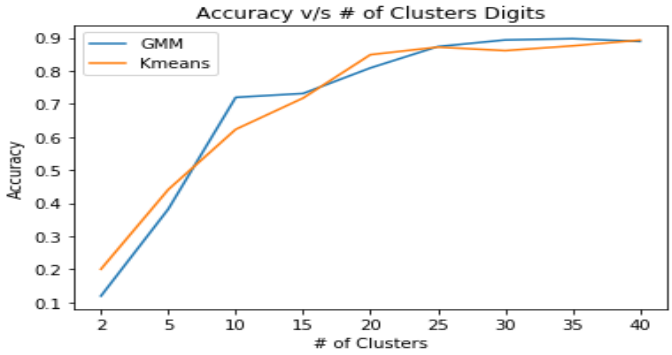


Comparing K-Means and EM

Below are the graphs of the accuracies obtained by both the algorithms on both the datasets. We can see that there is a sharp increase of accuracy for all the line plots. This is because beyond this point the model starts to overfit. Also we can notice that this steep increase ends when we reach the actual number of clusters in the dataset. This indicates that our algorithms perform well overall.

One more thing to notice is that K-means performs better for the Cancer dataset and EM performs better for the Digits dataset. This means that the Cancer dataset has a greater separability that is numerical. This means that most of the features are numerical in nature and thus can be separated by distance based clustering. On the other hand, Digits dataset does not only have quantifiable numeric features. Hence, EM performs better. (Makes sense because digit recognition should work even for digits with a slightly different orientation and length, as long as the digit is the same.)





Dimensionality Reduction (Section 3)

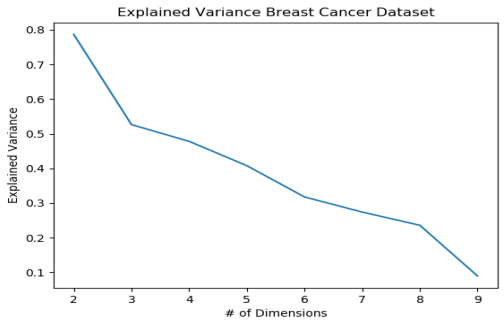
It refers to reducing the number of dimensions in a given dataset such that we do not loose significant information and can make accurate predictions. It is important in the real world because it helps save space required to store data and also significantly reduces the computational time. There are several dimensionality reduction algorithms available out of which I am going to analyze four. They are shown below.

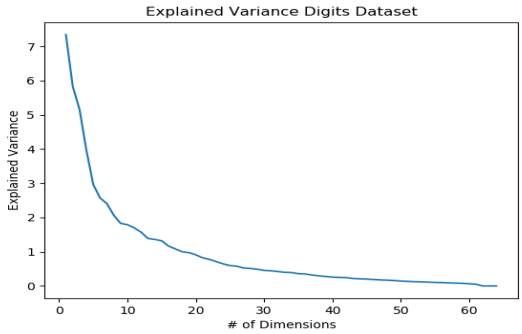
Principal Component Analysis

PCA is a dimensionality reduction technique that works by creating new dimensions from a given large set of dimensions. The new dimensions (also known as Principal Components) area linear combination of the original large set of dimensions. Thus, in this way they end up preserving all the information that the dimensions provide. Ideally, the principal components try to explain the variance in the dataset. So the first principal component puts the maximum effort at doing that. The second principal component tries to explain the variance that the first one couldn't and so on.

The below two graphs show the explained variance of the principal components. The explained variance is nothing but the ratio of an eigenvalue and the total variance. So effectively, the explained variance helps us in analyzing the eigenvalue better.

We know that each principal component tries to encapsulate the information about the variance of the initial dataset. And every consecutive PC tries to incorporate the variance knowledge that could not be incorporated by the PC's before it. This way we can say that when the number of PC's = no. of data points, the variance will be equal to 0. Thus, for every increment in the number of dimensions, the explained variance decreases and thus the model becomes better. In order to find the best number of dimensions to choose based on this graph, we should select a point such that the line before that is a steep curve and the line after that is a slowly decreasing curve. (For eg, in the digits dataset, that would be no. of dimensions = 10)

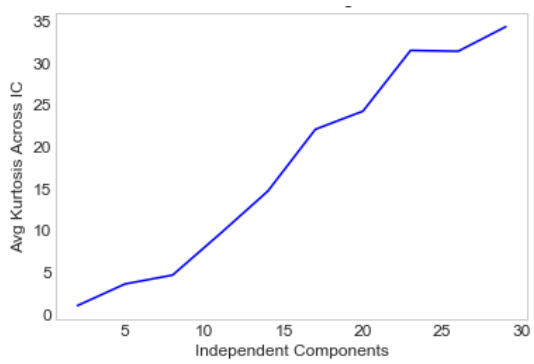


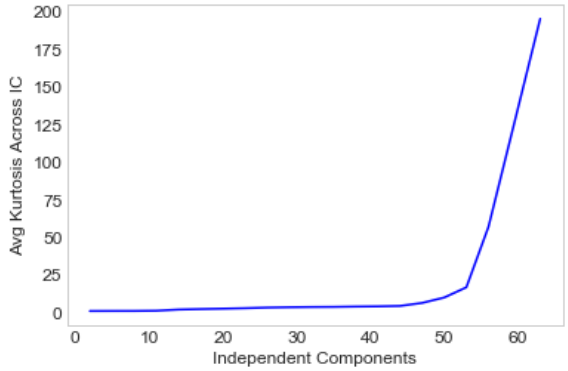


Independent Component Analysis (ICA)

This algorithm performs dimensionality reduction by looking for independent factors, as opposed to PCA that looks for uncorrelated factors. What this means is that this algorithm assumes that the given variables (dimensions) are a linear mixture of some other (lesser in number) variables. In addition, it also assumes that these new variables are mutually independent.

Below, I have plotted the graphs for the variation in the average kurtosis with the number of independent components (Cancer data on the left and Digits data on the right). Kurtosis is a measure of shape. Maximizing the kurtosis will make the distribution Non-Gaussian and thus help in getting independent components. Now, this logic directly coincides with the fact that in both the graphs below, an increase in kurtosis means an increase in the number of independent components. In order to get an effective solution, we should opt for a high peak within the range of our preference to get the best results in that range. Another important advantage of this algorithm is that it will help eliminate the overlapping data points that have not been dealt with until now.

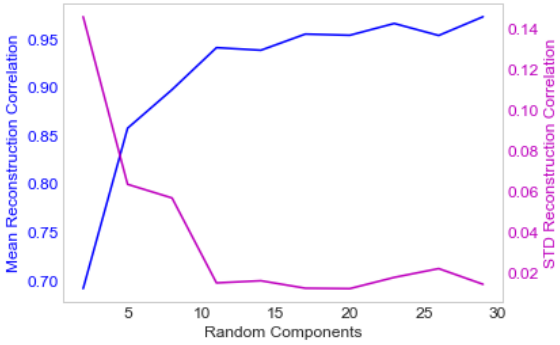


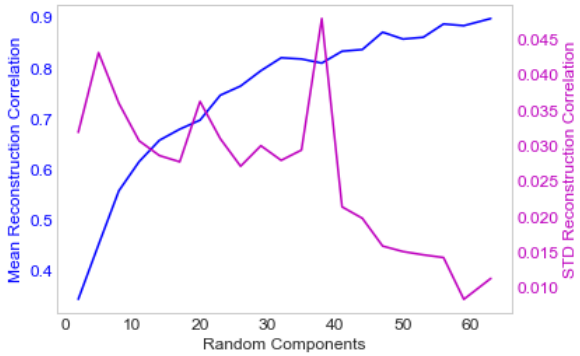


Randomized Projections (RP)

This method makes use of the lemma that given a set of points in a high dimension, we can project it to a space of lower dimension such that the distances between them are preserved. So, we can say that this is a dimensionality reduction method that aims to reduce the dimensions in a Euclidian space.

Based on the above description of the algorithm, we can assume that if our dataset had all numeric attributes then in that case, this algorithm would perform better. Below are the plots of the performance of this algorithm on both the datasets (left is for Cancer and right is for Digits). We know from previous analysis that the cancer dataset has numeric attributes and the results from this analysis further strengthen that conclusion (as the curves are steeper and reach the local optimum far earlier than that for the Digits dataset). Also, we can observe that there is a huge fluctuation in the curves for the Digits dataset. This is because the algorithm is inherently randomized and the clusters are quite closely spaced in the Digits dataset and thus the algorithm is not converging as quickly as it did for the Cancer dataset.

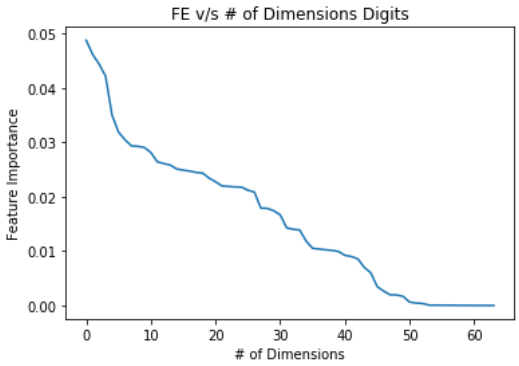


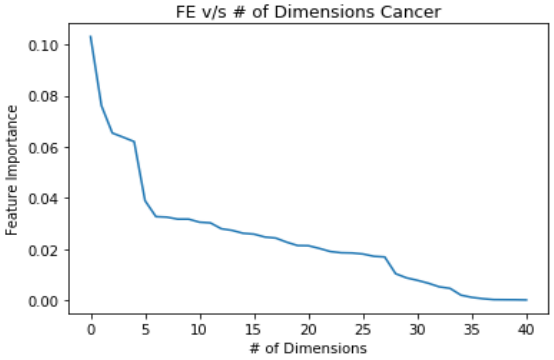


Random Forest (RF)

I chose Random Forest as my fourth feature selection algorithm because till now all the algorithms have focused on selection either based on the correlation, independence or distance and including Random Forest will give a fourth different flavor to my analysis. With RF, I am to decipher ordinal aspects of the data and analyze the performance based on them. Random Forest is basically bagging applied to a Decision Tree (in the sense that it is an average of multiple Decision Trees). Random Forest is a feature selection algorithm that makes the choice based on feature importance. This requires that all the input variables have only numerical values. Both my datasets fit this requirement, thus making RF a good feature selection algorithm to test on the data.

Below is the plot for the variation in feature importance with respect to the number of dimensions. We can see that for the Cancer dataset, the downward slope is very steep as compared to the Digits dataset and also that the initial value of feature importance of the Cancer dataset is higher. This is because there is less redundancy in the features of the Cancer dataset as compared to the Digits dataset. This means that each feature within the Digits dataset does not provide as much information as the features of the Cancer dataset. This makes sense because the features of the digit dataset are just values of the 8x8 bins. So adjacent bins would have similar values and thus contribute towards the redundancy.

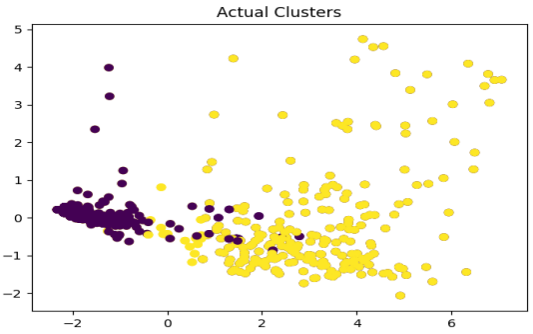


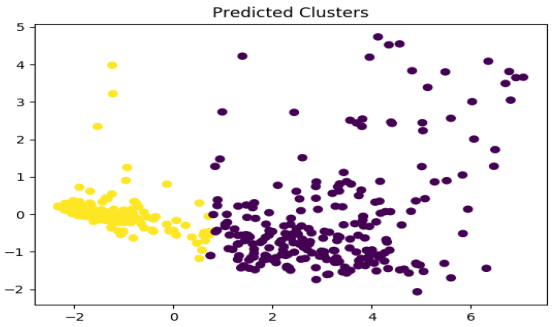


Clustering with Dimensionality Reduction (Section 4)

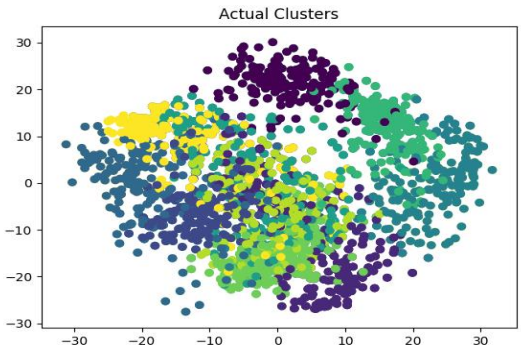
Clustering with PCA

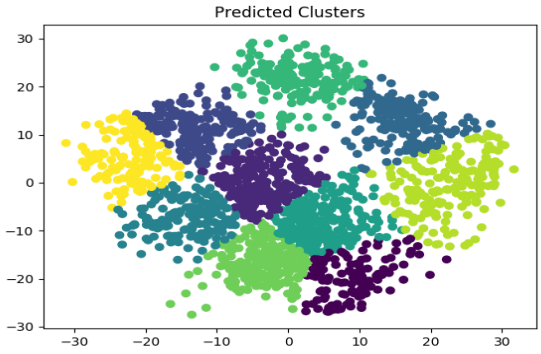
I performed PCA on both the datasets and plotted the clusters after the reduction. In the cancer dataset we can see that although the dimensions have significantly reduced, the overlap of the data points in the actual clusters still persists and our prediction, although correct overall, cannot incorporate that effectively.





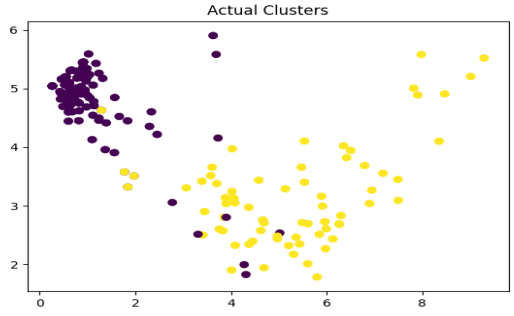
A similar trend can be noticed for the Digits dataset too. Even though the number of data points has reduced significantly, PCA cannot successfully remove the overlapping data points (which are quite likely outliers). Despite of that, the overall clustering performance seems to have improved from the previous section. This is because even though some of the points are incorrectly grouped, this algorithm has taken significantly lesser computational power to run, making it better overall.

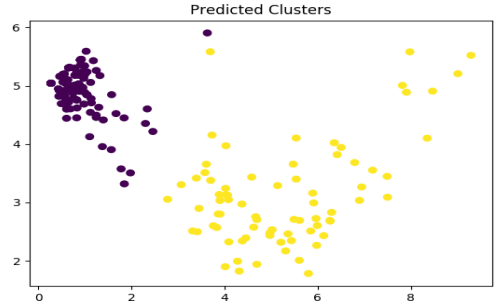




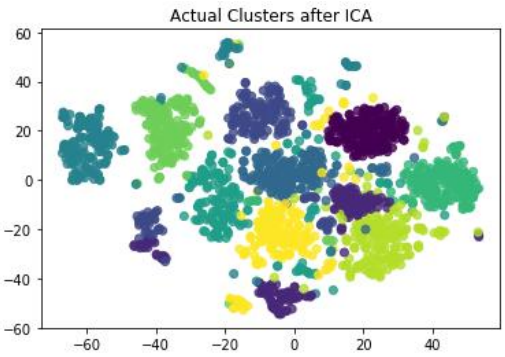
Clustering with ICA

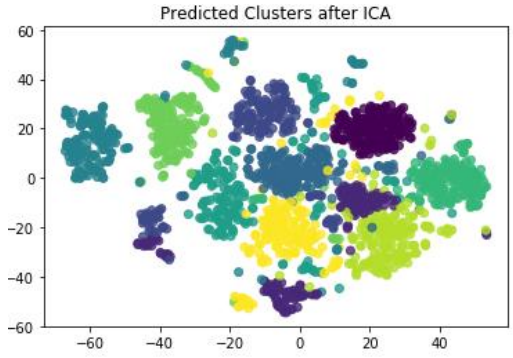
I performed ICA on both the datasets and plotted the clusters after the reduction. In the cancer dataset we can see that the overlap of the data points has reduced. This makes sense because inherently ICA aims at finding the independent components. Also, we can see that a lot of the sparsely places data points have been included by the ICA and this is indicative of its mutual independence property.





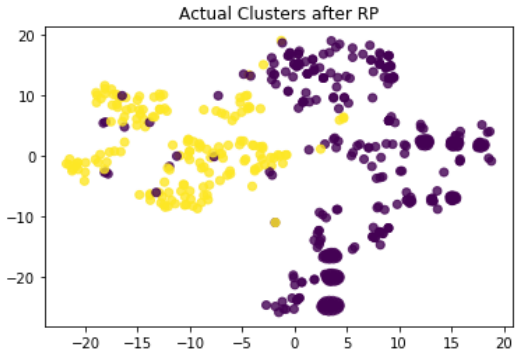
As for the digits dataset, we can see that the clusters are now spaced closer to each other with lesser overlap than PCA. This in turn makes it better for the algorithm to segregate the clusters more accurately. Thus, overall ICA seems to show a better performance than PCA. Despite of the better performance, once issue that persists in ICA with the digits dataset is that the clusters are spaced way closer to each other than with PCA.

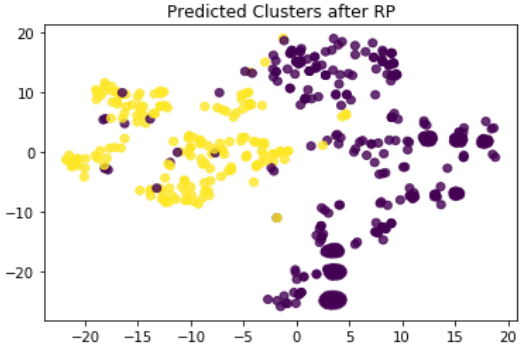




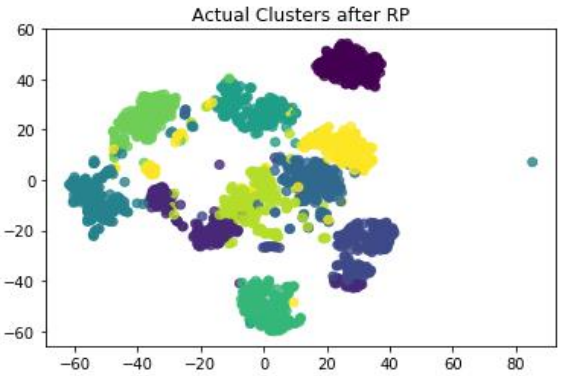
Clustering with RP

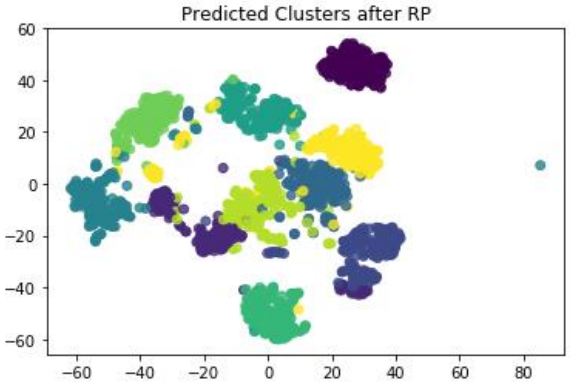
I performed RP on both the datasets and plotted the clusters after the reduction. In the cancer dataset we can see that the data points still overlap indicating that this dimensionality reduction technique does not work well with closely spaced data points. At the same time, the algorithm manages to segregate the greater chunk of the cluster accurately.





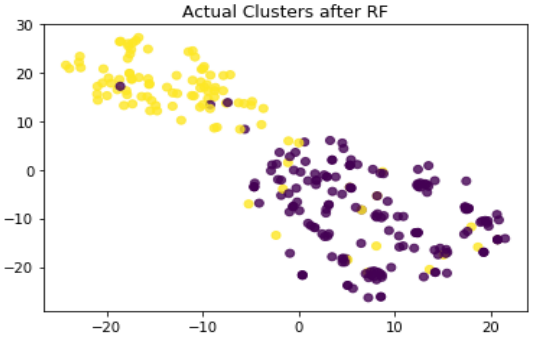
As for the digits dataset, the overlap of the data points is significantly lesser than that for the cancer dataset. This is because RP deals better with numeric data and since the digits dataset has ordinal values too, it cannot significantly reduce the number of dimensions while preserving the underlying feature information.

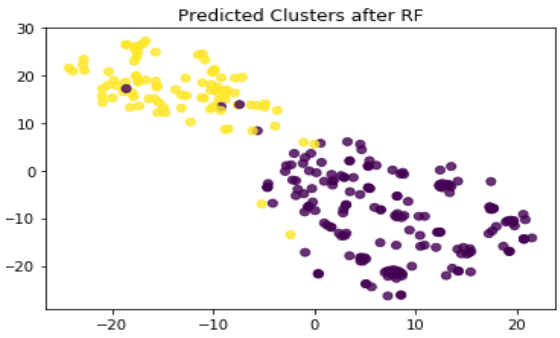




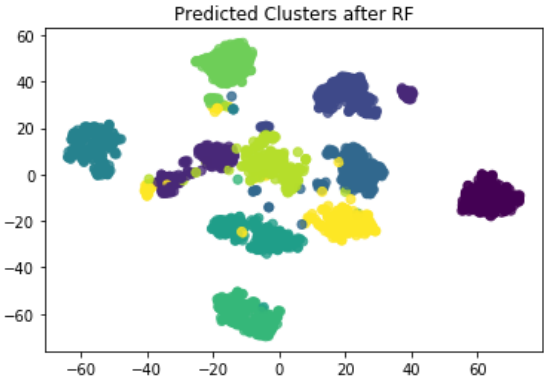
Clustering with RF

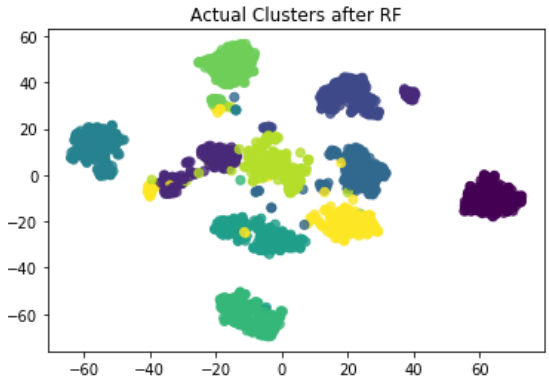
I performed RF on both the datasets and plotted the clusters after the reduction. In the cancer dataset we can see that this just like RP does not good job of eliminating the overlapping points. Despite of this, it does capture the sparsity of the overall cluster, which makes sense because RF reduces dimensions based on feature importance.





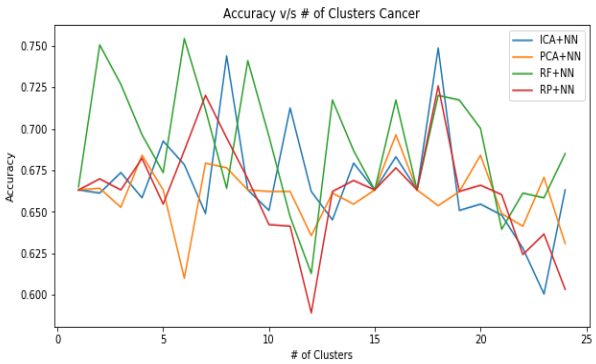
As for the digits dataset, we can see the numbers of overlapping points are lesser than produced by RP. This makes sense because RP is meant for numeric data whereas this reduces the dimensions based on feature importance. This is also the reason why the clusters are spaced apart (as a decision tree like structure would split the clusters based on hard threshold).

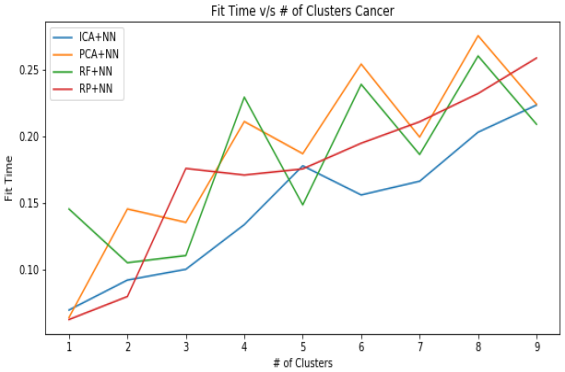




Neural Network on the Cancer Data after Dimensionality Reduction (Section 5)

For this section, I perform the analysis on the dataset used from assignment 1 (Cancer). Earlier on using the neural network on this dataset, I achieved an accuracy of 74.3% and the mean fit time was approximately 0.7. But checking the performance of neural network, after applying dimensionality reduction techniques has shown an increase in the performance by giving an accuracy of 75.4%. This seems counterintuitive as we are removing several dimensions but since the dataset is wide to begin with, applying dimensionality reduction techniques vastly reduces the chance of overfitting. At the same time, since we are dealing with lesser dimensions, the mean fit time is also significantly reduced to 0.2 (approx.).



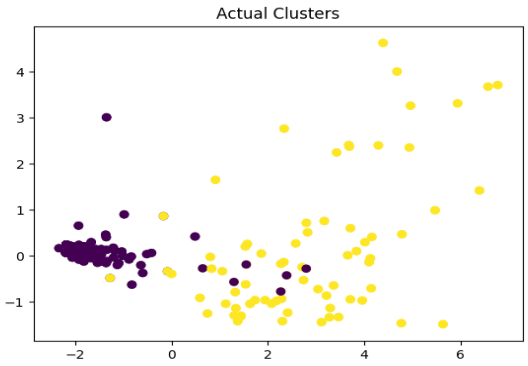


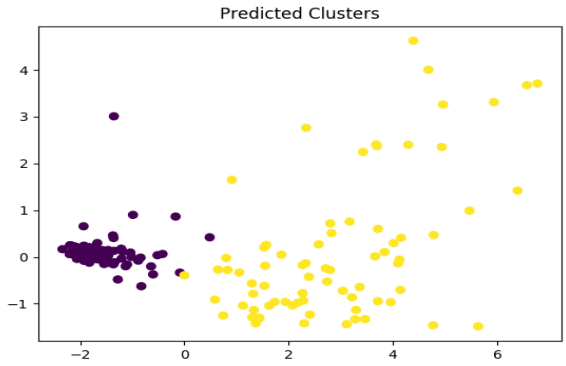
Clustering as Dimensionality Reduction and Neural NN on Cancer Dataset (Section 6)

In this section, I used clustering algorithms as a replacement of dimensionality reduction algorithms to be applied to the neural network from assignment 1. On doing so, I observed that using clustering gave better results than a normal neural network (the reason for this is similar to the one mentioned above). Amongst the two clustering algorithms, K-means performs better than EM. This is because the underlying dataset (cancer) has numerical features and K-means being distance based clustering algorithms is more fit for this problem than the probability based clustering algorithm. The observations are as follows:

Algorithm	Time (Speed)	Performance
Normal NN	1.57	73.0248
K-Means + NN	0.21	75.8964
EM + NN	0.26	75.0235

Since K-means perform the best, I decided to plot the data points and see the clusters formed by using K-means + NN. We can see that the actual data still has some overlapping data points which are incorrectly predicted by the algorithm but the non-overlapping, sparse data points are accurately predicted which is an improvement over the previous iteration.





Conclusion

In conclusion, of the two clustering algorithms analyzed above, K-means works well when the given feature space can be segregated based on distance (Cancer dataset). On the other hand, EM works well when the given dataset can be segregated based on probability (Digits dataset). As for the dimensionality reduction algorithms, each of them performs differently based on the given problem (explanations provided above) and should be used after understanding the problem. Finally, for the cancer dataset, applying clustering and dimensionality reduction algorithms prior to the neural network have shown better performance and speed as opposed to just using the neural network.

References

1) Jonathan Tay - Github

2) UCI ML Repository- Cancer Dataset