Unsupervised Learning and Dimensionality Reduction

Josh Adams
jadams334@gatech.edu

INTRODUCTION

I will be using two different datasets, the MNIST and the Fashion-MNIST. Both datasets contain 60,000 training examples and 10,000 testing examples. The two datasets are interesting in many ways, the first would pertain to the MNIST dataset which contains handwritten digits from 0 to 9. When one person writes a 5 it can and typically does look different than a 5 written by a different person, this adds variability to the dataset for all the digits. The fashion-MNIST dataset contains images of different types of clothing each with many different shapes and many different intensities.

PART - CLUSTERING

Clustering data using k-means clustering and expectation maximization. Both are attempting to find underlying structure in the dataset and have different means of obtaining it. The first aspect needed to be tackled was if preprocessed made a difference in the results. I will use inertia and average silhouette score to compare scaled and not scale datasets. The inertia is a measurement of the sum of square distances between the points and the centroid of the cluster. The higher the inertia the more spread the cluster. I chose to scale my datasets using a min-max scaler, which moved my dataset ranges from 0 to 255 to 0 to 1. I came to this conclusion after many tests which the results are included in the provided container folder. I will not include the results as the would take up valuable report space. The reason behind the scaling was due to the values my dataset features being 0 to 255 and the number of features. This would create an extremely large dimensional space which would not work well for distance clustering, such as K-means clustering.

1.1 K-Means Clustering

K means cluster is an unsupervised learning technique which attempts to find underlying structure in the data. It does this by establishing k clusters and assigning the observations to those clusters. It will then calculate the centroid of those clusters, move the center of the cluster to the centroid and redesignate observations to the newly positioned clusters. I used the elbow method along with a silhouette score based on correlation to identify a good number of clusters to use. The elbow is a coarser evaluation based on the rate of change of the distortion. After a certain number of clusters, the benefit of adding more diminishes. I charted the correlation of the distortion as the number of clusters increased.

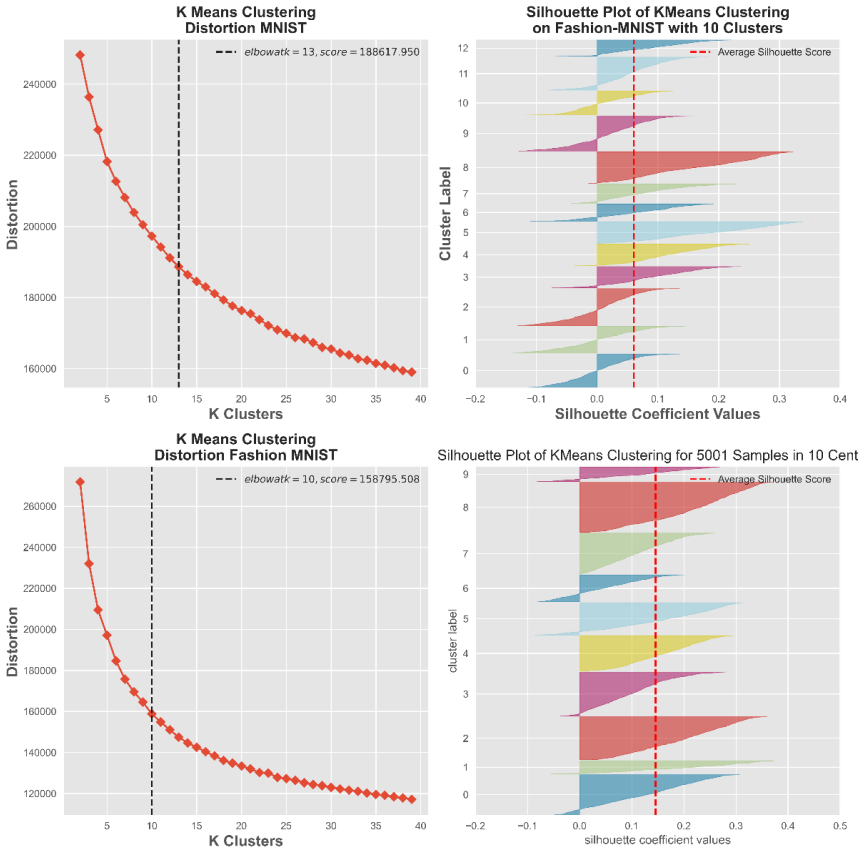

Figure 1 — Elbow method K Means Clustering with Silhouette.

Looking at the silhouette scores for both datasets, would suggest that 13 and 10 clusters would be a good option as there is little negative silhouette scores and almost all of the clusters have an average silhouette score greater than the average.

1.2 Expectation Maximization

Expectation maximization was another clustering technique used. Comparison of the AIC and BIC were used to determine an appropriate number of components. Typically, you want the lowest AIC and the highest BIC, for the MNIST dataset using 10 components was ideal and for the Fashion-MNIST dataset it appeared to be closer to 9 components.

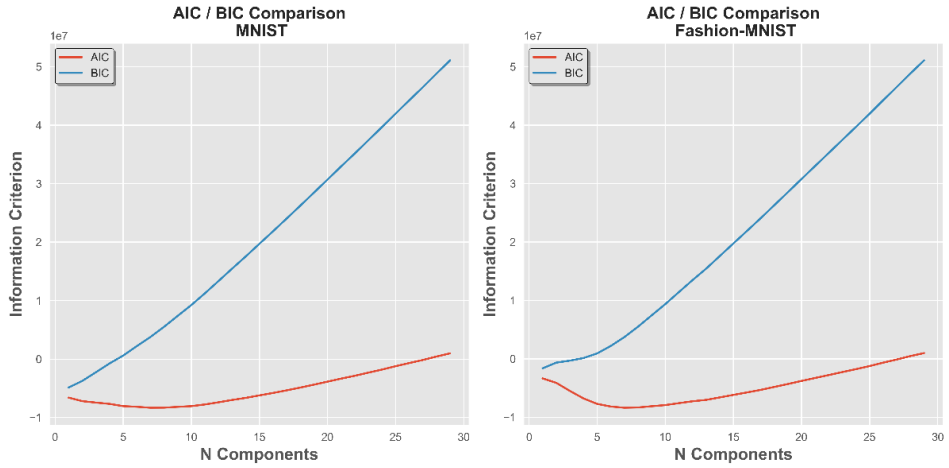

Figure 2 — AIC/BIC Expectation Maximization MNIST and Fashion-MNIST.

PART – DIMENSIONALITY REDUCTION

Applying 4 different dimensionality reduction techniques to our datasets. Principal Component Analysis (PCA), Independent Component Analysis (ICA), Randomized Projections and Random Forest Classifier.

1.3 Principal Component Analysis

I wanted to capture the most variance while using the least number of features. The MNIST dataset I selected 326 allowed for a significant reduction in features while still able to account for 0.99% of the variance found in the data. The eigen values for both the MNIST and Fashion-MNIST dataset are next to their charts.

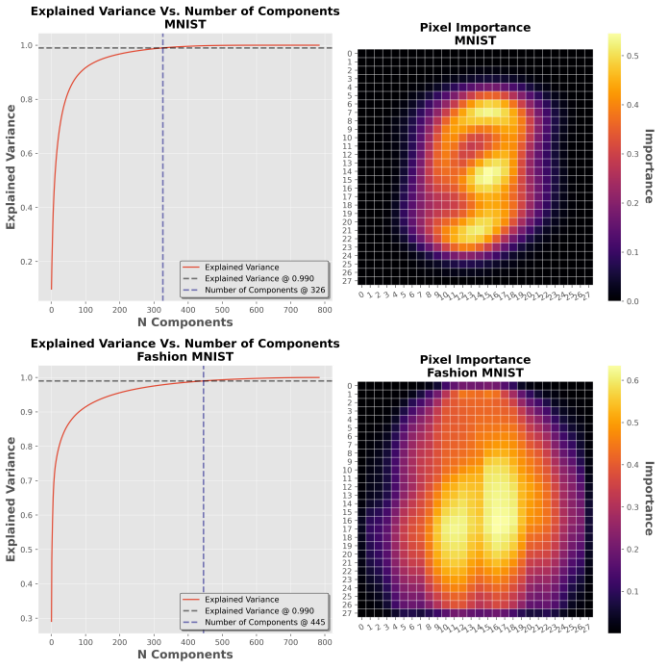

Figure 3 — Principle Component Analysis dimension reduction on both datasets.

The results of mapping out the eigen values uncovers more about the datasets. The MNIST dataset contains almost most of its information in a small area in the center of the image. The Fashion-MNIST dataset eigenvalues, which is considered a more complicated dataset, show that its data is much more spread which alludes to it being more complicated, but also the limitations of how many features can be thrown away. The MNIST dataset has many more erroneous features as evident by the larger black spaces in the image.

1.4 Independent Component Analysis

Kurtosis for both datasets have been plotted along with its pixel importance and reconstruction error. Having a higher kurtosis when dealing with ICA is preferable but alone does not provide enough information to make a good decision about the dimension reduction.

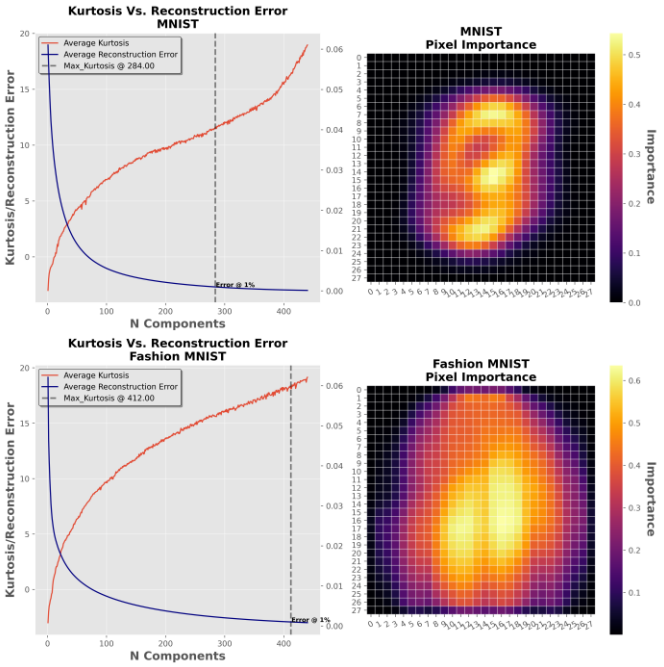

Figure 4 — Independent Component Analysis dimension reduction on both datasets.

I then plotted the reconstruction error so I would be able to take the number of components with the highest kurtosis while maintaining accurate data reconstruction. The MNIST dataset was able to be reduced to 284 features and the Fashion-MNIST to 412 features, both are significant reductions in feature space from the original datasets 784 features.

1.5 Randomized Projections

Randomized projection is the more difficult algorithm to work with. I plotted the reconstruction error which was my main method for determining the number of features able to be reduced. To maintain a reconstruction error of at most 10%, 715 features and 746 features were chosen for the MNIST and the Fashion-MNIST datasets, respectively.

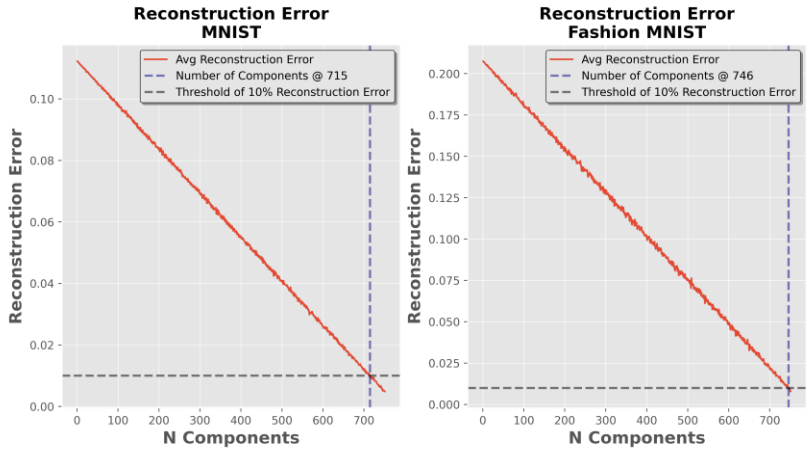

Figure 5 — Randomized Projection dimensionality reduction on both datasets.

Since this was a stochastic based algorithm, I ran it 20 times for every number of components and averaged the results.

1.6 Random Forest

Random forest was my favorite to work with because how simplistic it was, the large number of features it was able to reduce and its overall accuracy. How this works is you train a forest of decision tree classifiers on the dataset and then average the feature the trees split on over the entire forest. This results in the significance levels of each feature. I was not sure of an accurate way to calculate reconstruction error. I decided to use 3 other classifiers to gauge the reconstruction accuracy of the random forest. For each number of components, I selected the N most important and ran a decision tree, an SVM and KNN on the resulting reduced dataset. I chose the decision tree because the forest is comprised of thousands of them and I wanted to verify my methods were working as intended.

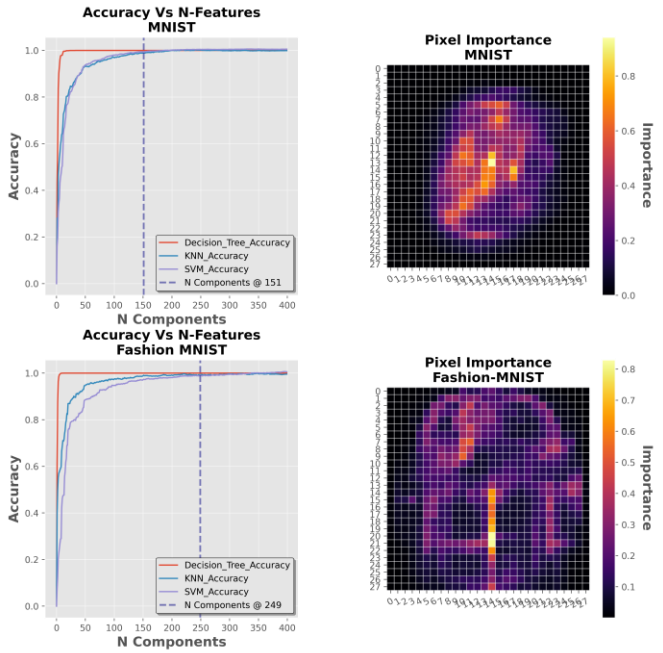

Figure 6 — Random Forest dimensionality reduction on both datasets.

The SVM was used because the goal is reducing dimension but an SVM works by projecting the data into higher dimensional spaces. I the reduced dataset was able to be projected back into a larger dimensional space then I would expect the reduction is at the least maintaining the datasets integrity. KNN was used because it is a distance-based classifier and if we are reducing the number of features then the distances between observations is lowering, which would make it '*hopefully*' easier for the algorithm. If KNN performed poorly this would suggest I either needed to scale my data or my dimension reduction technique was not working as I expected.

Table 1 — MNIST Dimensionality Reduction Results

Name	N-Components	Reduction Pct	Reconstruction Accuracy
PCA	*326*	0.58 %	0.99 %
ICA	*284*	0.63 %	0.99 %
Random Projections	715	0.08 %	0.90 %
Random Forest	152	0.81 %	0.99 %

Table 2 — Fashion-MNIST Dimensionality Reduction Results

Name	N-Components	Reduction Pct	Reconstruction Accuracy
PCA	445	0.43 %	0.99 %
ICA	412	0.47 %	0.99 %
Random Projections	746	0.05 %	0.90 %
Random Forest	250	0.68 %	0.99 %

PART – REPRODUCE CLUSTERING EXPERIMENTS

Run both K-Means and Expectation Maximization on the dimensionality reduced datasets.

1.7 Principal Component Analysis

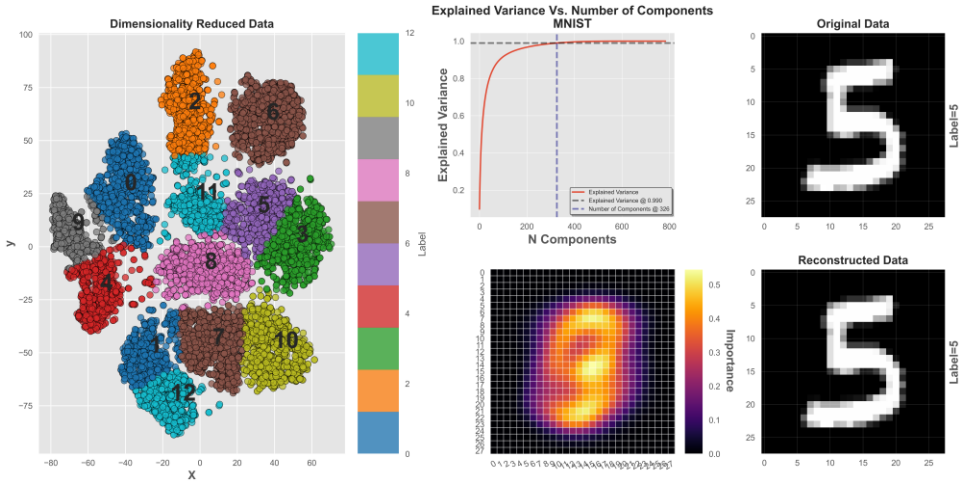

Figure 7 — Principal Component Analysis pair plot on MNIST dataset.

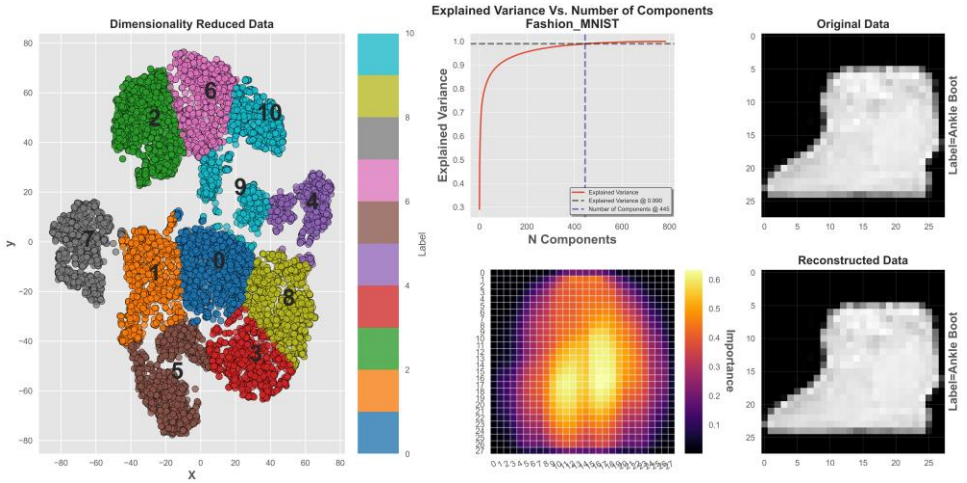

Figure 8 — Principal Component Analysis pair plot on Fashion-MNIST dataset.

Some interesting aspects about the clustering on the PCA reduced dataset would be how the clusters were divided. I would have combined the 9th cluster and the 4th and many others.

1.8 Independent Component Analysis

ICA produced some of the more interesting clusters. The first thing to notice is the larger number of clusters used. The next and in my opinion most interesting aspect of the clustering was how most clusters are grouped together and there is a segment of empty space then a *'wall'* of outliers —most notably on the Fashion-MNIST dataset—.

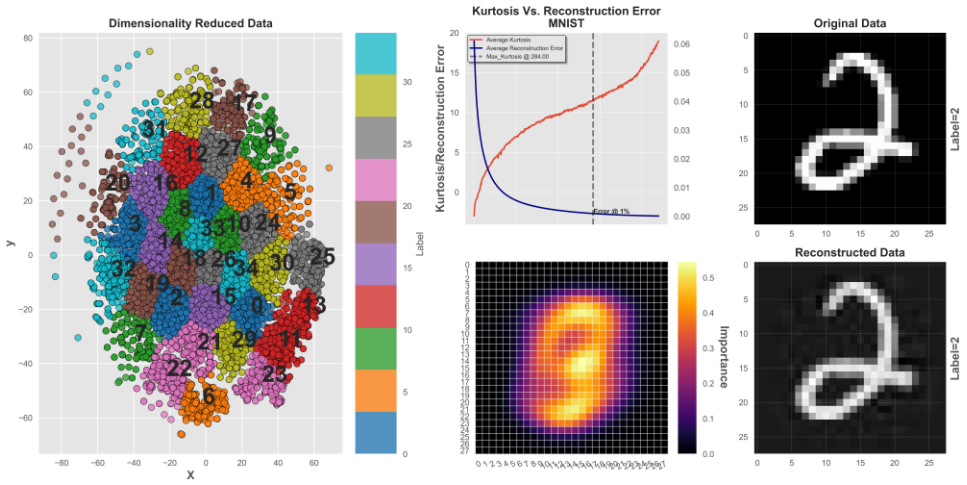

Figure 9—Independent Component Analysis pair plot on MNIST dataset.

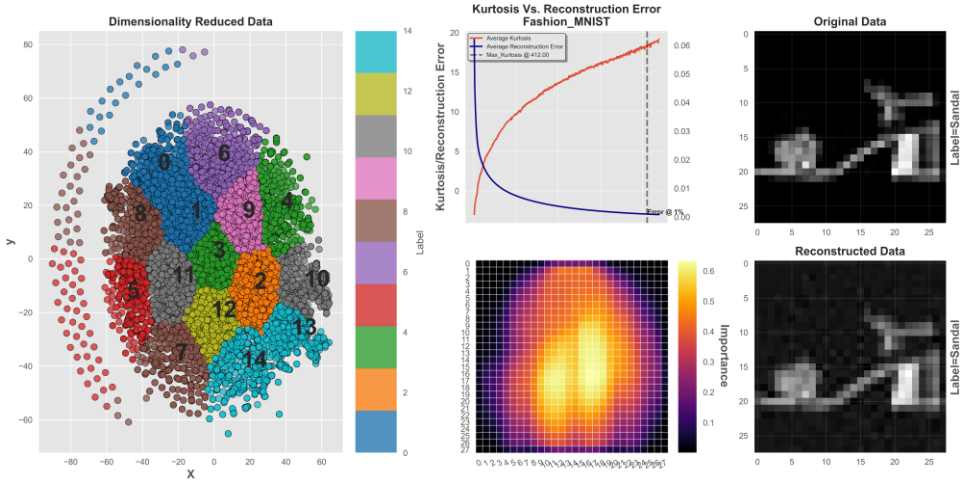

Figure 10—Independent Component Analysis pair plot on Fashion-MNIST dataset.

1.9 Randomized Projections

I did not find the clustering of random projections to be particularly interesting, however I found the reconstruction to be fascinating. If you look at the reconstruction of the original, you can easily decipher what the image is of. The major difference between random projections and the other dimensionality reduction techniques was the amount of noise introduced to the data. The reconstruction on both datasets used over 700 of the 784 features, I would have expected the reconstruction to be much better than it was.

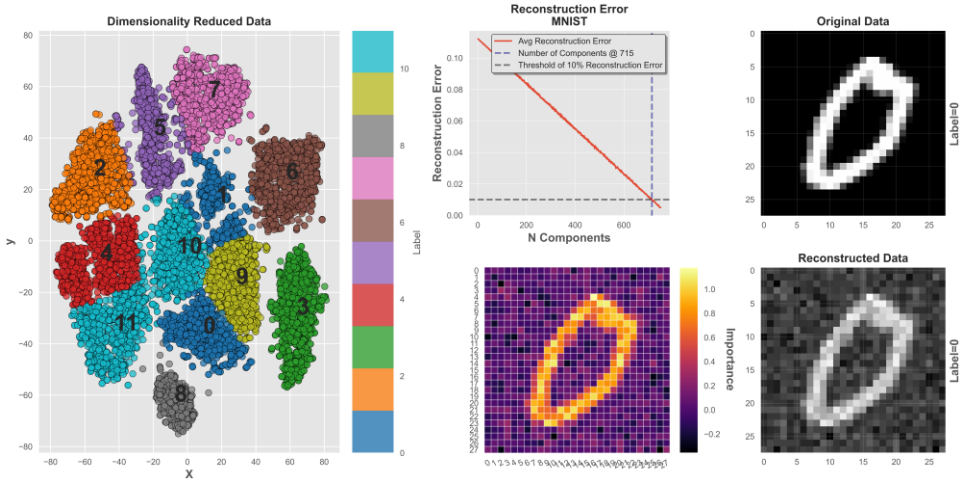

Figure 11—Randomized Projections pair plot on MNIST dataset.

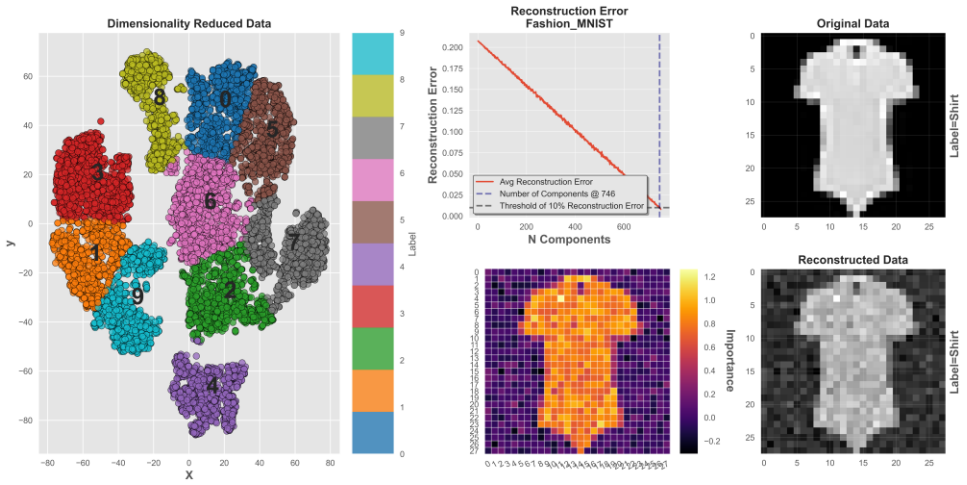

Figure 12 — Randomized Projections pair plot on Fashion-MNIST dataset.

1.10 Random Forest

Using the random forest as a dimensionality reduction technique worked very well. It was able to significantly reduce the number of features in the dataset while preserving most of the data —shown in the reconstructions—

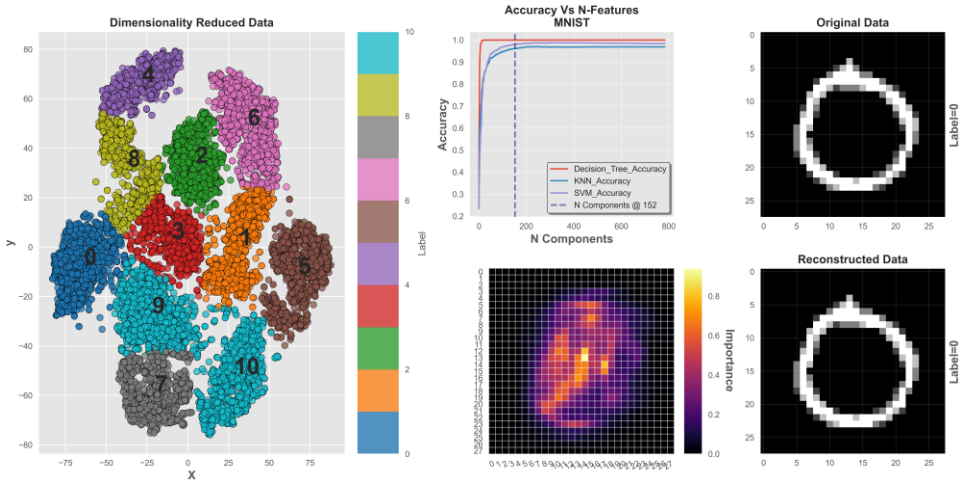

Figure 13 — Random Forest pair plot on MNIST dataset.

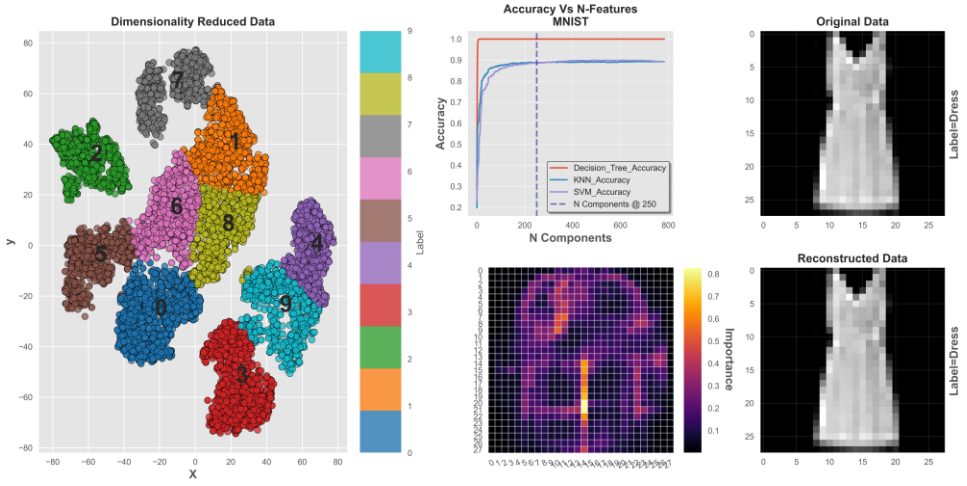

Figure 14 — Random Forest pair plot on Fashion-MNIST dataset.

PART – NEURAL NETWORK ON DIMENSIONALITY REDUCED DATASET

Using various dimensionality reduction techniques on the Fashion-MNIST dataset to reduce the number of features from 784 to some number based on the evaluations of the reduction technique. The dimensionality reduction techniques used will be PCA, ICA, Randomized Projections and Random Forest from the previous sections. PCA reduced

the number of features to 445, ICA to 412, Randomized Projections to 746 and Random Forest to 250. PCA, ICA and Random Forest were able to preserve as much as 99% of the information while significantly reducing the number of features. Randomized projections did not perform well and was extremely time consuming.

Each algorithm was able to reduce the number of features but in many cases had trade-offs which needed to be considered. There was a distinct tradeoff that was discovered was the increase to training time even though the number of features was lowered.

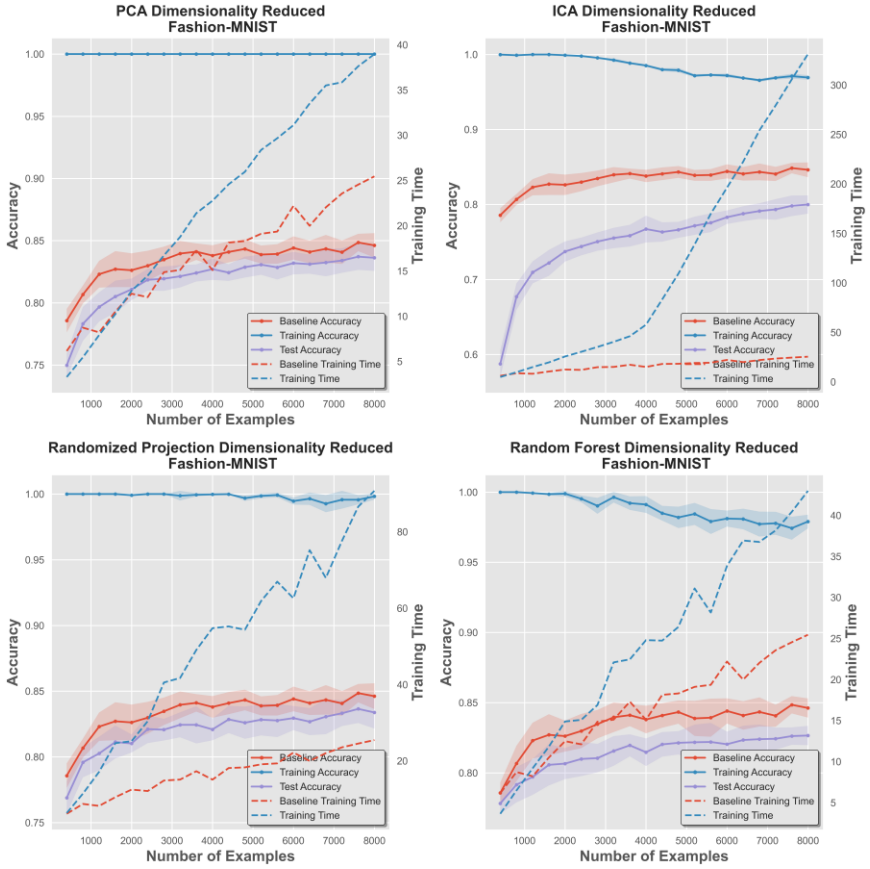

Figure 15—Neural Network on dimensionality reduced Fashion-MNIST dataset.

PCA was almost able to achieve the performance of the baseline while using many fewer features. The almost doubling in training time for PCA, which was verified three times, is something which should be considered. ICA testing accuracy was able to approach the baseline and I do think it would have either met or surpassed with enough training examples. The big difference is the training times being over 10x the baselines. I could not determine the cause, but I believe it was due to the scaling the reduced data. I was not able to confirm my thoughts because of time constraints. Randomized projections performed well on the test set and was within 2 to 4 percent of the baseline. The training time for randomized projections was almost three times that of the baseline. Random forest was the best in my opinion because it was comparable to PCA in terms of training time, while using almost half as many features as PCA. The random forest showed that it also was reducing overfitting because the training accuracy was going down and the testing accuracy was increasing. With more observations the training and testing accuracy would eventually converge to a point and reduce no further due to irreducible error in the data.

Table 3 — Neural network results on DR datasets, 8000 observations.

Name	N Features	Training Accuracy	Testing Accuracy	Training Time (s)
Baseline	*784*	*1.0*	*0.845*	*25.0*
PCA	*445*	*1.0*	*0.840*	*40.0*

Name	N Features	Training Accuracy	Testing Accuracy	Training Time (s)
ICA	*412*	*0.98*	*0.802*	*340.0*
Random Projections	*746*	*0.99*	*0.841*	*93.0*
Random Forest	*250*	*0.98*	*0.83*	*44.0*

PART – CLUSTERING AS DIMENSIONALITY REDUCTION

Clustering provides new data in the form of the clusters, which may be explored. This part of the assignment we will used the newly created clusters as features in a dataset to hopefully gain more knowledge about our dataset and possible ways to improve our performance. Using both K-Means clustering and Expectation Maximization (EM) to cluster the Fashion-MNIST dataset as a means of dimensionality reduction take slightly different routes to achieve the goal of dimensionality reduction. K-Means clustering will take our features space of 784 and group them into N clusters and this will leave you with a new feature space of just those clusters. EM takes the dataset and rather than clustering the data explicitly, it produces a prediction of what cluster an observation belongs to. Once the prediction array is obtained, apply one hot encoding to the array to produce a new larger array where each value for the cluster prediction is treated as a feature.

For both algorithms I compared their training and testing accuracy against the baseline, which was the default neural network used in previous assignments and the entire feature space of 784. The training times for baseline and clustered algorithms are included. Both K-Means and EM performed much better than I had expected. K-Means was able to average above 70% accuracy on both training and testing set, this is impressive as only 11 features were used. This would be over a 99% reduction in number of features while being less than 20% lower in accuracy. EM found 15 clusters this would also reduce the number of features by 99% and was still able to maintain an accuracy of around 60%. I do believe that this would be the best performance I could achieve with these new datasets. The reason is that both their training and testing accuracies quick converge and stagnate. What would be beneficial would be to either use more clusters or adding these newly generated features to an existing dataset. I believe appending this reduced dataset to other dimensionality reduced datasets would be a much better approach because you are still able to maintain the benefits of lower dimensionality while augmenting the datasets with more information.

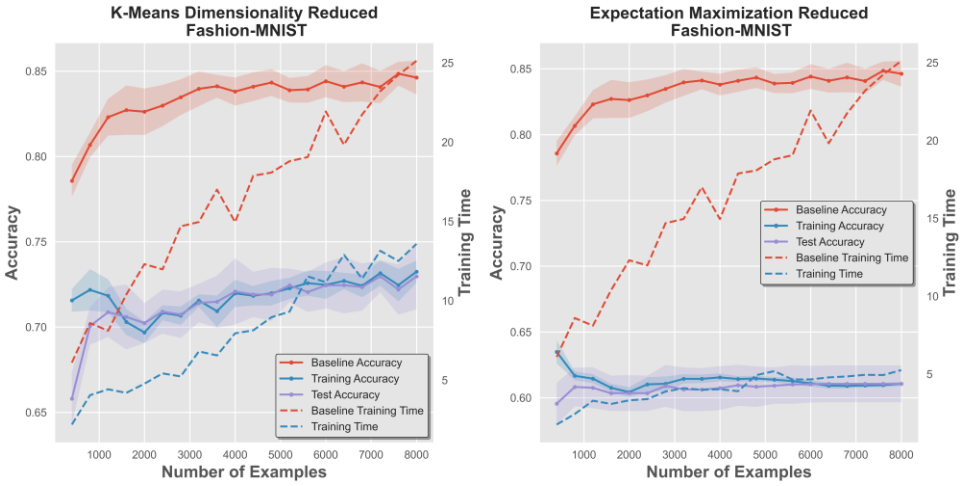

Figure 16—Clustering as dimensionality reduction technique.

The training time for the k-means reduced dataset grew at approximately the same rate as the original dataset which was interesting. I had expected the training time to be much faster as the feature space was drastically reduced. EM performed more how I had expected because it was using much less data. The training time of EM for 2000 observations was very close to the training time of 8000 observations. The reason I believe the two algorithms training times were so different was due to the features we used in the dimensionality reduced dataset. For k-means we have 11 features but each of those features are continuous values from 0 to 20. EM used 15 features but instead of being continuous values they were essentially discretized due to the one hot encoding. This means that for each observation all features are zero except for

one and that value is limited to only the whole numbers in the range 0 to N features. What this means is that while k-means has less features, its feature space is vastly larger than the feature space found with EM. I was not able to perform these experiments due to time constraints.

REFERENCES

1. LeCun, Yann, et al. "THE MNIST DATABASE." *MNIST Handwritten Digit Database, Yann LeCun, Corinna Cortes and Chris Burges*, Nov. 1998, yann.lecun.com/exdb/mnist/.
2. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, 86(11):2278-2324, November 1998.
3. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. Han Xiao, Kashif Rasul, Roland Vollgraf.
4. https://zhiyzuo.github.io/EM/
5. https://www.scikit-yb.org/en/latest/index.html
6. https://scikit-learn.org/stable/
7. https://matplotlib.org/
8. https://pandas.pydata.org/
9. https://numpy.org/
10. https://www.scikit-yb.org/en/latest/
11. https://1drv.ms/u/s!AvQKh521Q29Jg_tVfoK5AUF1GmTrOw?e=Yo8wSY
12. https://www.dropbox.com/s/0kg02s3xzv4gy8x/UL-Container.zip?dl=0