CS 7641 Assignment 3: Unsupervised Learning and Dimensionality Reduction

Boyuan Liu
bryanliu@gatech.edu

1. Introduction

Clustering is one of the most common tasks in unsupervised learning. Clustering algorithms such as K-Means and Expectation Maximization can group data points together based on their similarity in features. Dimensionality reduction is also a common technique in machine learning. It uses linear transformations to reduce the dimensionality of the features while preserving information. In this assignment, I explored the K-Means and Expectation Maximization clustering algorithms and dimensionality reduction methods. The transformed dataset was then tested with neural networks to study the impact of clustering and dimensionality reduction.

2. Dataset

The datasets used in this assignment were the UCI Handwritten Digits dataset and the Credit Approval dataset [1], the same as Assignment 1. The Handwritten Digits dataset contains 1797 images of handwritten digits. Each image is 8x8 pixels in size. The most common digit is 3 with 183 samples and the least common digit is 8 with 174 samples [2]. The Credit Approval Dataset dataset has a total of 690 entries. The data has 15 categories including age, gender, marital status, education level, etc. The target is a boolean value of whether the application for credit was approved or not. 44.5% of the applications were denied and 55.5% of the applications were approved [2].

3. Clustering

In this section, I explored the K-Means and Expectation Maximization clustering algorithms in an unsupervised way, i.e., not looking at the labels. I calculated the Within Cluster Sum of Squares (WCSS) and used the ‘elbow’ method to determine the number of clusters for the K-Means algorithm. The ‘elbow’ method aims to find a model that has a low WCSS with relatively fewer clusters. For the Expectation Maximization algorithm, I used the Bayesian information criterion (BIC) as the primary metric. The BIC was calculated for models with different numbers of clusters and types of the covariance

matrix. The covariance matrix types include full, tied, diagonal, and spherical. The model with the lowest BIC score was selected. According to the WCSS and BIC scores of the Credit Approval dataset shown in Figure 1, the 'elbow' was at approximately 10 clusters and the model with the lowest BIC score was marked with '*'. Therefore, 10 clusters were chosen for K-Means and 9 clusters with the diagonal covariance matrix were chosen for EM. The label distribution in each cluster for each algorithm is also shown in Figure 1. The EM algorithm was able to separate the data points with different labels fairly well, especially in clusters 3, 5, and 6. The K-Means algorithm, on the other hand, did not separate the data as well as EM. Some clusters contain fairly mixed labels, such as cluster 0, 2 and 8, which means the clusters did not align very well with the labels. The K-Means algorithm could potentially be improved by further hyperparameter tunings, such as the relative tolerance and the number of initialization.

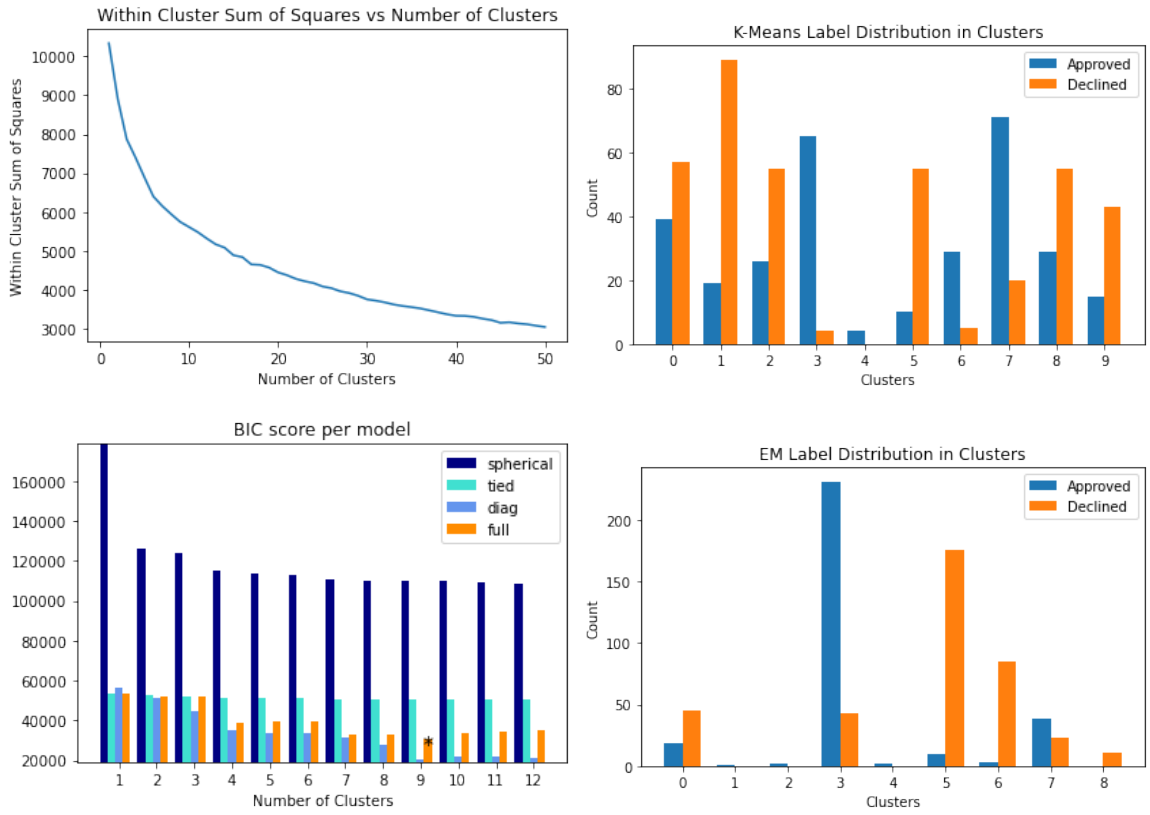


Figure 1 - K-Means and Expectation Maximization clustering on the Credit Approval dataset.

In the Pen Digits dataset, I used the same methods to determine the number of clusters and covariance matrix types for K-Means and EM. As shown in Figure 2, the 'elbow' was at approximately 15 clusters and the model with the lowest BIC score was marked with '*'. Therefore, 15 clusters were chosen for K-Means and 17 clusters with the diagonal

covariance matrix were chosen for EM. The label distribution within clusters of K-Means shows that most of the clusters were dominated by one label, which means the clusters aligned very well with the labels. The label distribution in EM clusters shows that the clusters also aligned well with the labels. Although in several clusters, it had slightly more noise than K-Means.

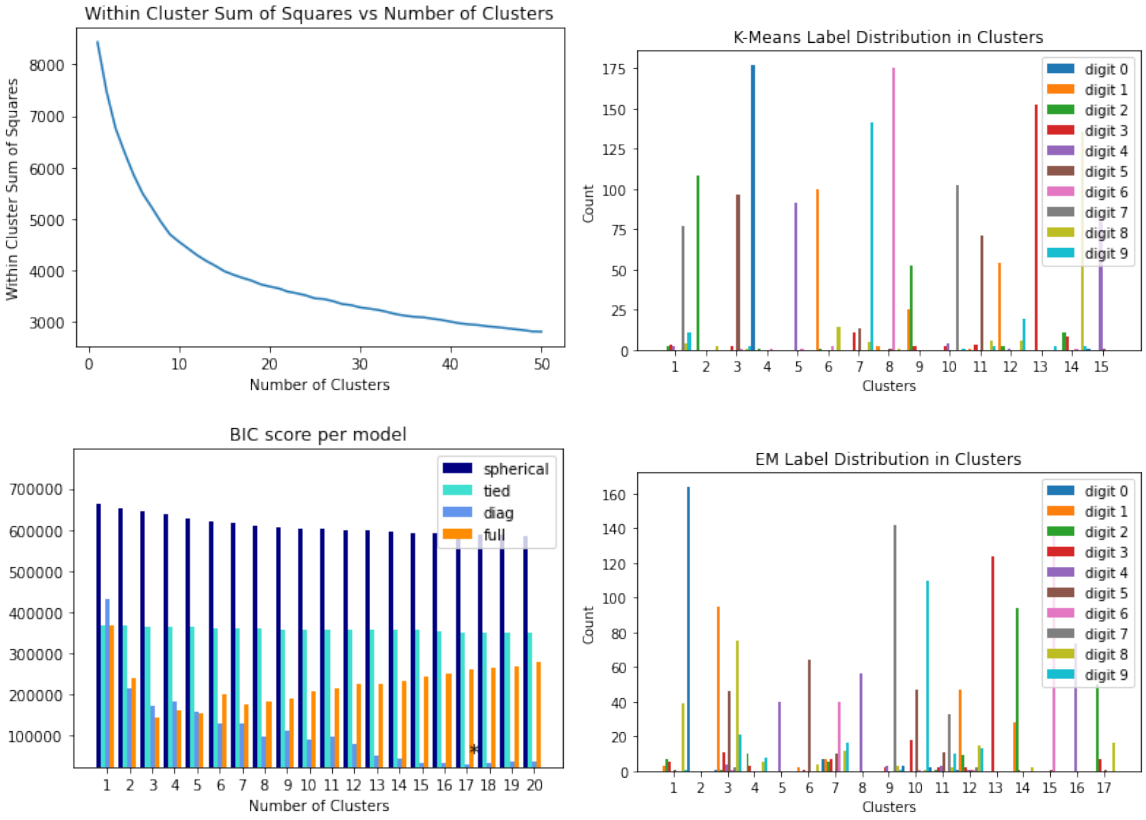


Figure 2 - K-Means and Expectation Maximization clustering on the Pen Digits dataset.

4. Dimensionality Reduction

Dimensionality reduction uses linear transformations to reduce the dimensionality of the features while preserving information. In this section, I explored four dimensionality reduction algorithms: PCA, ICA, Randomized Projection, and Truncated SVD. Each algorithm has its own criteria to determine the optimal number of components. For PCA and SVD, the distribution of eigenvalues is obtained through the explained variance attribute. The optimal solution aims to have low explained variance while keeping the number of components relatively low as well. 11 components were chosen for both algorithms. For ICA, the kurtosis value was computed for each component. Eight components were chosen to achieve the maximum Kurtosis value while keeping the number of components low. For randomized projection, the reconstruction error is the

primary metric. The algorithm was executed ten times to better illustrate its general trend in reconstruction error. Seven components were chosen in order to minimize the reconstruction error.

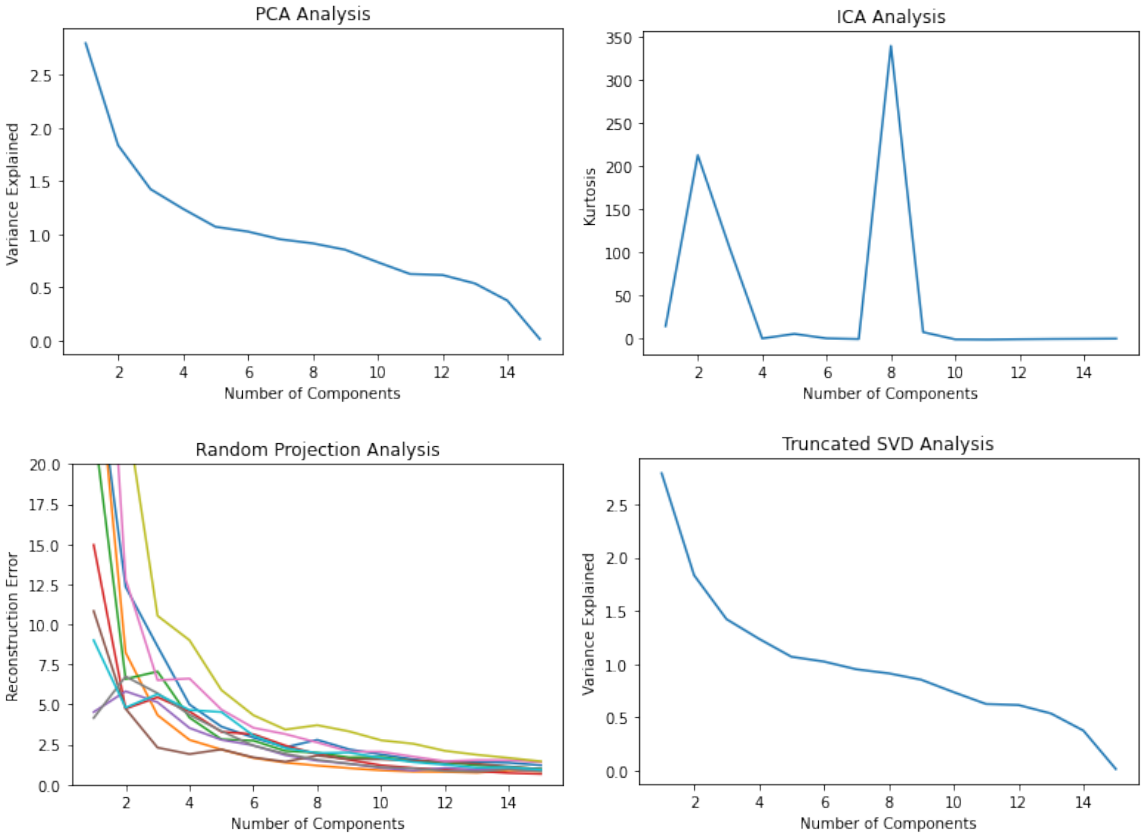
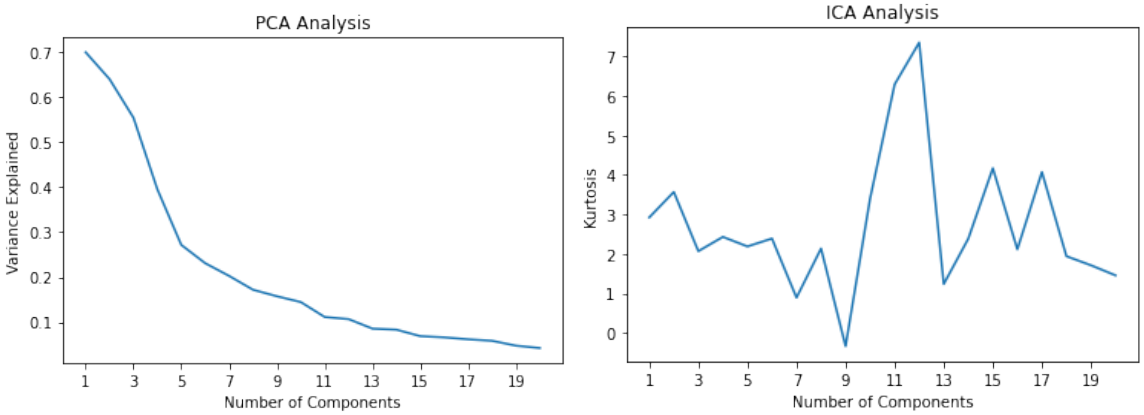


Figure 3 - Analysis of dimensionality reduction algorithms on the Credit Approval dataset.

For the Pen Digits dataset, the number of components for each algorithm was determined with the same methods. As shown in Figure 4, the metrics show a similar trend. The number of components was 11, 12, 8, and 11 for PCA, ICA, RP, and SVD, respectively.



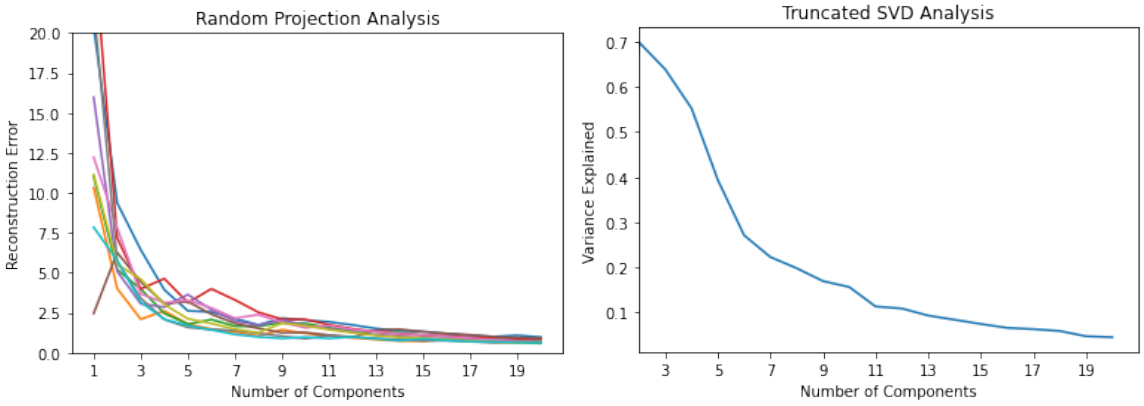
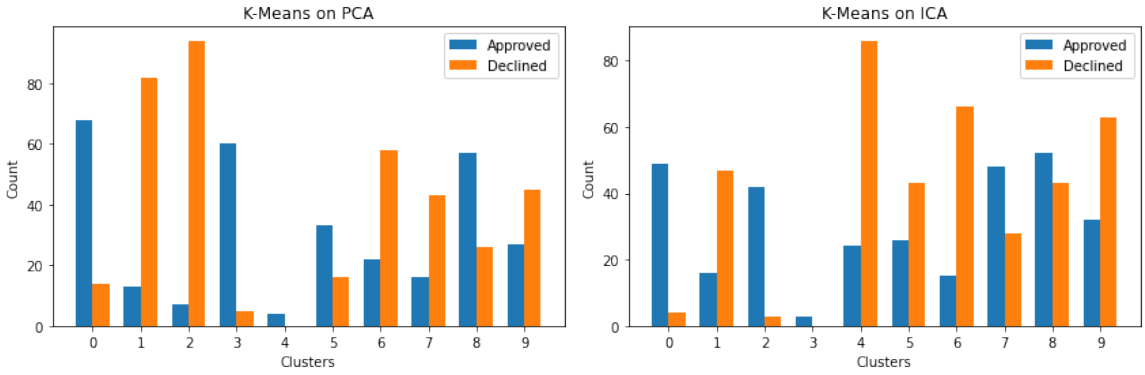


Figure 4 - Analysis of dimensionality reduction algorithms on the Pen Digits dataset.

5. Clustering on Dimensionality Reduced Dataset

After obtaining the dimensionality-reduced datasets, I applied K-Means and EM clustering to the datasets and examined their performances. The label distribution within clusters for the Credit Approval dataset is shown in Figure 5. K-Means performed well on PCA and SVD data. In the PCA data, cluster 0, 2, and 3 consist of mostly data points with the 'declined' labels and cluster 1, 4, and 5 consist of mostly data points with the 'approved' labels. In the SVD data, cluster 0, 3, and 4 consist of mostly data points with the 'declined' labels and cluster 1 and 2 consist of mostly data points with the 'approved' labels. K-Means did not perform very well on the Randomized Projection data. Three of the five clusters have almost evenly distributed labels, which indicates a poor separation of the data points. EM did not perform as well as K-Means in general on the dimensionality-reduced dataset. Many clusters contain very few data points, which are essentially outliers in the dataset. Nevertheless, EM was able to separate the PCA and SVD data into two major clusters that are dominated by different labels.



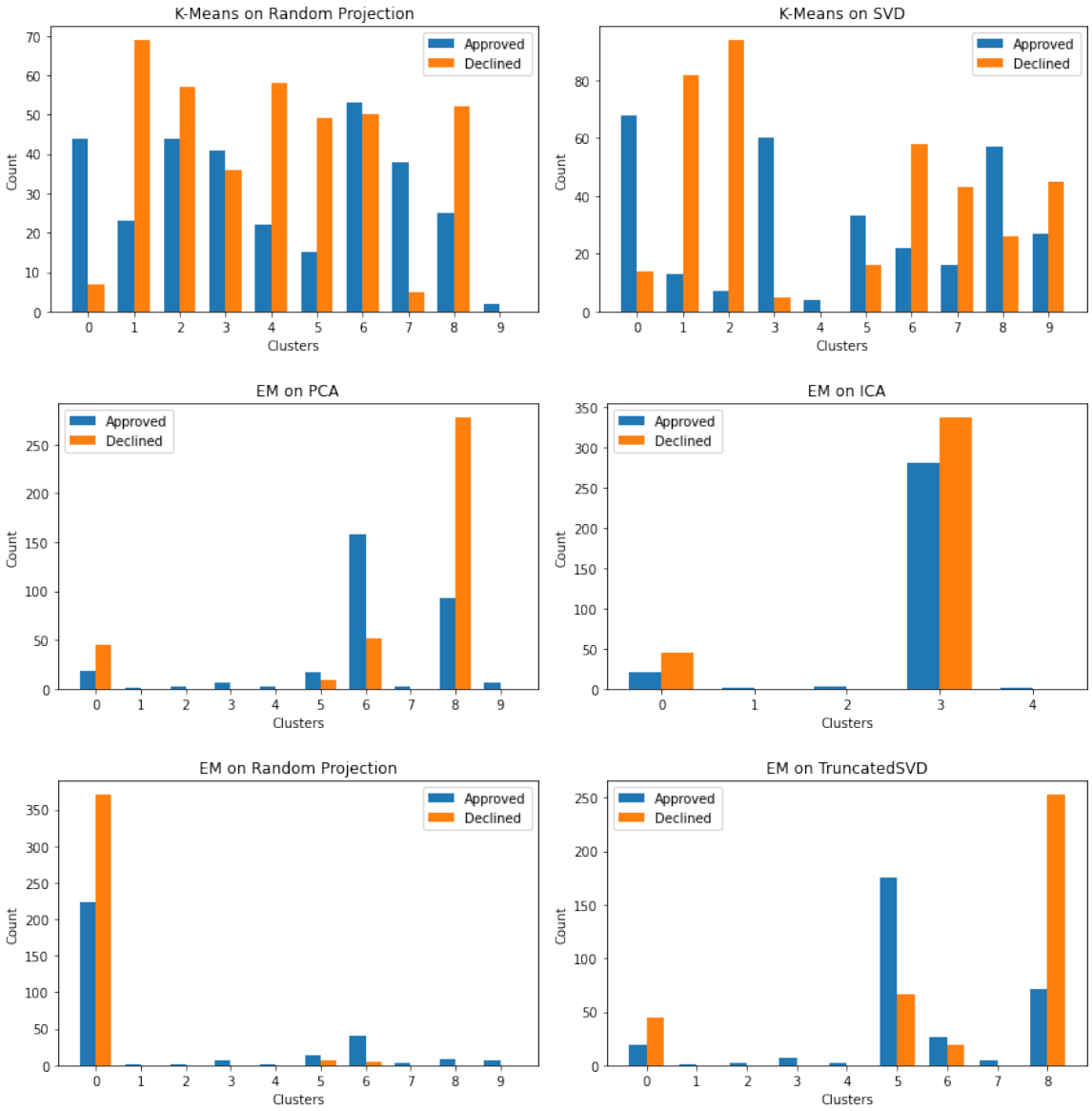


Figure 5 - Label distribution within clusters for the dimensionality reduced Credit Approval dataset.

Unlike the previous dataset, both K-Means and EM performed well on the dimensionality -reduced Pen Digits dataset. EM tends to divide the dataset into more clusters than K-Means. As shown in Figure 6, more clusters generally leads to better separation of the labels. On the other hand, more clusters could also lead to overfitting. With that kept in mind, K-Means on ICA and EM on PCA appear to have the best overall performance among the eight combinations.

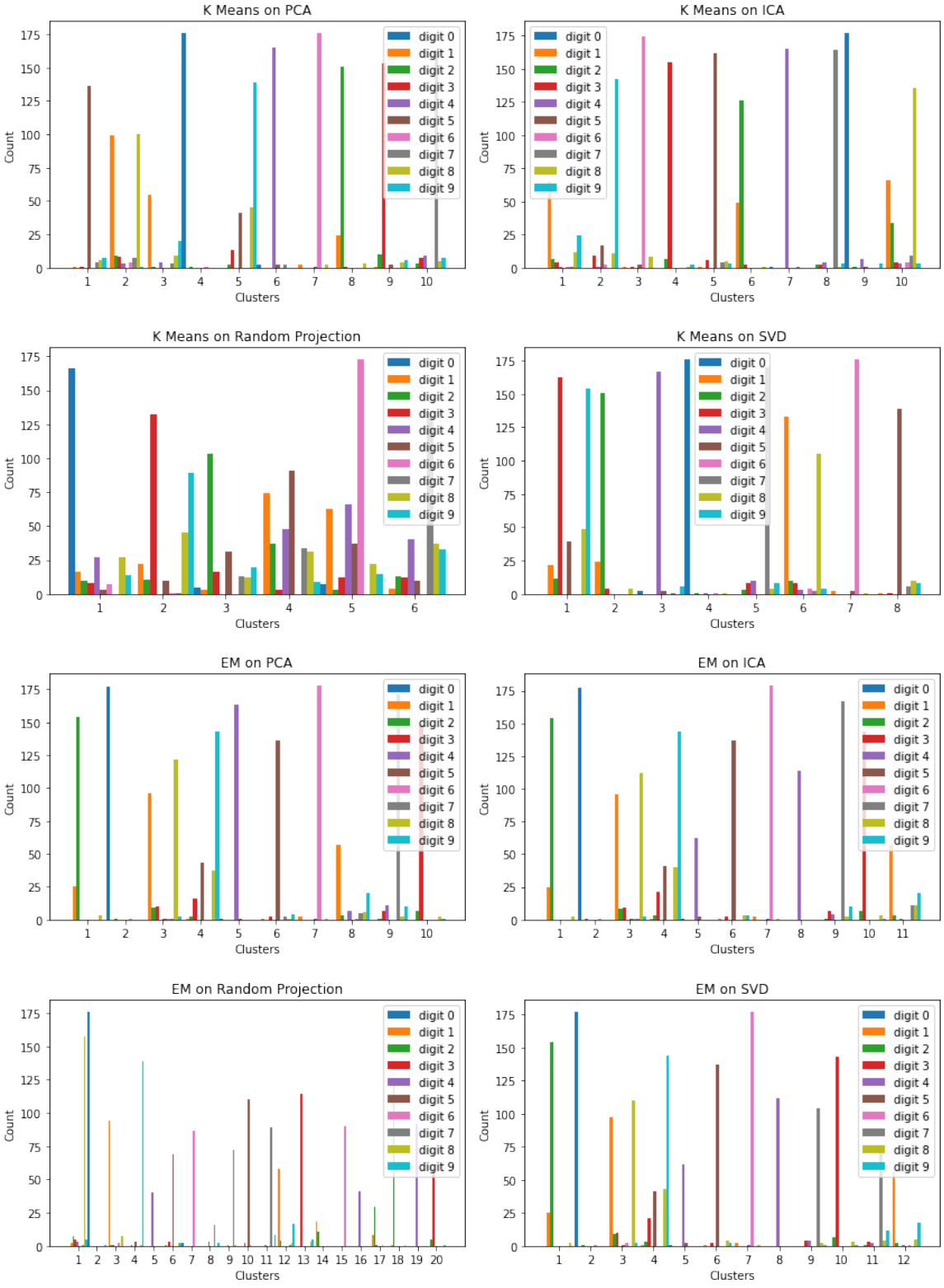


Figure 6 - Label distribution within clusters for the dimensionality reduced Pen Digits dataset.

6. Training Neural Networks with the Transformed Dataset

To further study the effectiveness of the transformed Credit Approval datasets, I applied MLP classifiers to the transformed datasets and compared their performances against the MLP classifier on the original dataset. The transformed datasets of PCA, ICA, Randomized Projection, and Truncated SVD have 11, 8, 7, and 11 features, respectively. I performed hyperparameter tuning on each neural network using the same methods as Assignment 1. The training results of five neural networks are shown in Figure 7. The training loss of PCA was very close to the original dataset, while the other three classifiers had slightly higher training losses. The testing accuracy and wall clock time of each classifier are shown in Table 1. The classifier on the original dataset achived the highest accuracy of 87.6%, while the other classifiers had slightly lower accuracies. This is because although the transformed datasets preserved the majority of the information, there was some distortion and information loss in the process. The classifier using the original dataset took the longest wall clock time. This is expected because the transformed datasets have more condensed features and therefore are faster to train on.

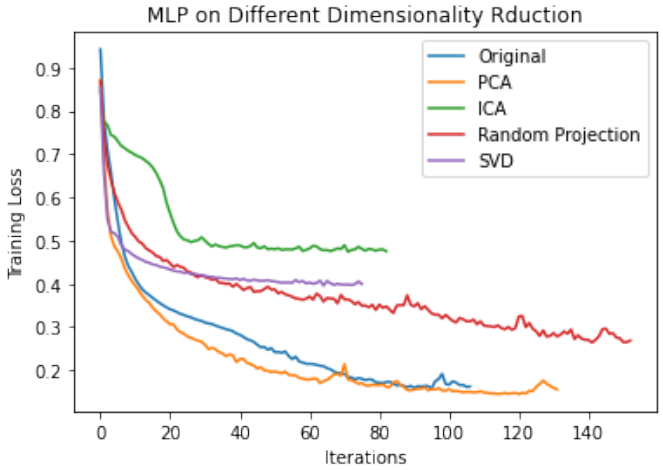


Figure 7 - Training loss of MLP classifiers using the transformed datasets.

Dataset	Test Accuracy	Time (s)
Original	0.876	0.35
PCA	0.833	0.30
ICA	0.840	0.17
Randomized Projection	0.804	0.34
SVD	0.826	0.16

Table 1 - Testing accuracy and wall clock time of the MLP classifiers.

7. Training Neural Networks with Additional Clustering Features

In the previous experiment, the classifiers using the transformed datasets did not perform as well as the classifier using the original dataset. To improve the transformed datasets, I applied the K-Means and EM algorithms to the datasets and used the cluster information as new features. The training results are shown in Figure 8. The loss curves of the classifiers show a similar trend to the previous experiment. ICA and SVD took more iterations to converge due to the additional features. The testing accuracy and wall clock time of each classifier is shown in Table 2. By comparing the results to the previous experiment, we can see that the additional clustering features increased the accuracy of PCA and SVD harmed the other two. This is because K-Means and EM only clustered the data well on the PCA and SVD datasets. The clusters generated on the other datasets contain more mixed data points, which is misleading information to the classifier. The classifiers already have over 80% accuracy without additional clustering features. In order to improve their performance with clustering information, the clusters need to have much higher purity. In terms of wall clock time, the classifiers on the transformed datasets had faster training time than the original dataset, similar to the previous experiment.

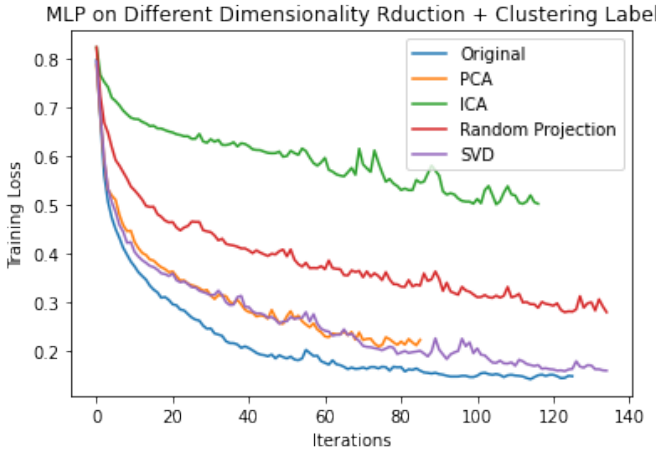


Figure 8 - Training loss of MLP classifiers using the transformed datasets + additional clustering features.

Dataset	**Test Accuracy**	**Time (s)**
Original	0.876	0.35
PCA	0.841	0.21
ICA	0.826	0.26
Randomized Projection	0.783	0.30

SVD	0.862	0.34

Table 2 - Testing accuracy and wall clock time of the MLP classifiers.

8. Conclusion

In this assignment, I explored the K-Means and Expectation Maximization clustering algorithm and four dimensionality reduction algorithms including the PCA, ICA, Randomized Projection, and SVD. The effectiveness of the transformed datasets was tested with MLP classifiers. The results have shown that although the MLP classifiers achieved good accuracies on the transformed datasets, they are not as effective as the original dataset in terms of accuracy. Feature transformation did reduce training time due to its more condensed feature information. The results have also shown that adding clustering information as additional features does not always help. The clustering information needs to be highly accurate to improve the classifier's accuracy.

Reference

[1] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Boyuan Liu. CS7641 Assignment 1: Supervised Learning. Georgia Institute of Technology. 2022.