Clustering Report

For this task, I trained two clustering models on the MNIST dataset: a k-means model and a gaussian mixture model.

To preprocess the data, I performed min-max scaling, mainly for numerical stability when calculating variance. In addition, clustering in a high-dimensional space is computationally expensive and susceptible to the Curse of Dimensionality, and so I applied PCA to reduce the number of features from 784 to 80. This captures the majority of the data's variance while making the clustering manageable, and I only fit PCA on the training data to prevent information leakage.

I chose optimal hyperparameters for each model by performing Grid Search Cross Validation.
For k-means, I explored these hyperparameters:

- n_clusters: [8, 10, 12, 15] - although there are 10 digits, testing higher numbers allows the model, for example, to split the "1"'s into slanted vs. straight, and so to possibly obtain better accuracy
- n_init: [10, 20, 30] – this controls the number of times the algorithm will run with different centroid seeds

For the gaussian mixture model, I explored:

- n_components: [8, 10, 12, 15] – the number of clusters, same reasoning as above
- covariance_type: ['full', 'tied', 'diag'] – full: each cluster has its own general covariance matrix, tied: all clusters share the same covariance, diag: eigenvalues of covariance are on the diagonal

I evaluated each model (trained with a permutation of the hyperparameters) with two metrics: Adjusted Rand Score (ARI) and V-Score.
ARI measures the similarity between the true labels and the predicted clusters, adjusted for chance. A score near 1.0 indicates near-perfect matching.
V-Score calculates the harmonic mean of homogeneity and completeness to evaluate how well predicted clusters map to true classes. A score of 1.0 indicates that clusters are both perfectly pure and comprehensive.
The final model was chosen based on the Average Score: (ARI + V-Score) / 2.

For k-means, the top 5 models were:

n_clusters	n_init	ARI	V-Score	Average Score
15	20	0.375607	0.535308	0.455458
15	30	0.374552	0.534597	0.454575
15	10	0.373774	0.533902	0.453838
8	10	0.395114	0.506208	0.450661
8	30	0.394930	0.505898	0.450414

For Gaussian Mixture Model, the top 5 models were:

n_components	covariance_type	ARI	V-Score	Average Score
10	full	0.395552	0.569897	0.482725
12	full	0.369077	0.575560	0.472318
15	full	0.361571	0.573687	0.467629
8	full	0.307010	0.526462	0.416736
10	tied	0.309473	0.472149	0.390811

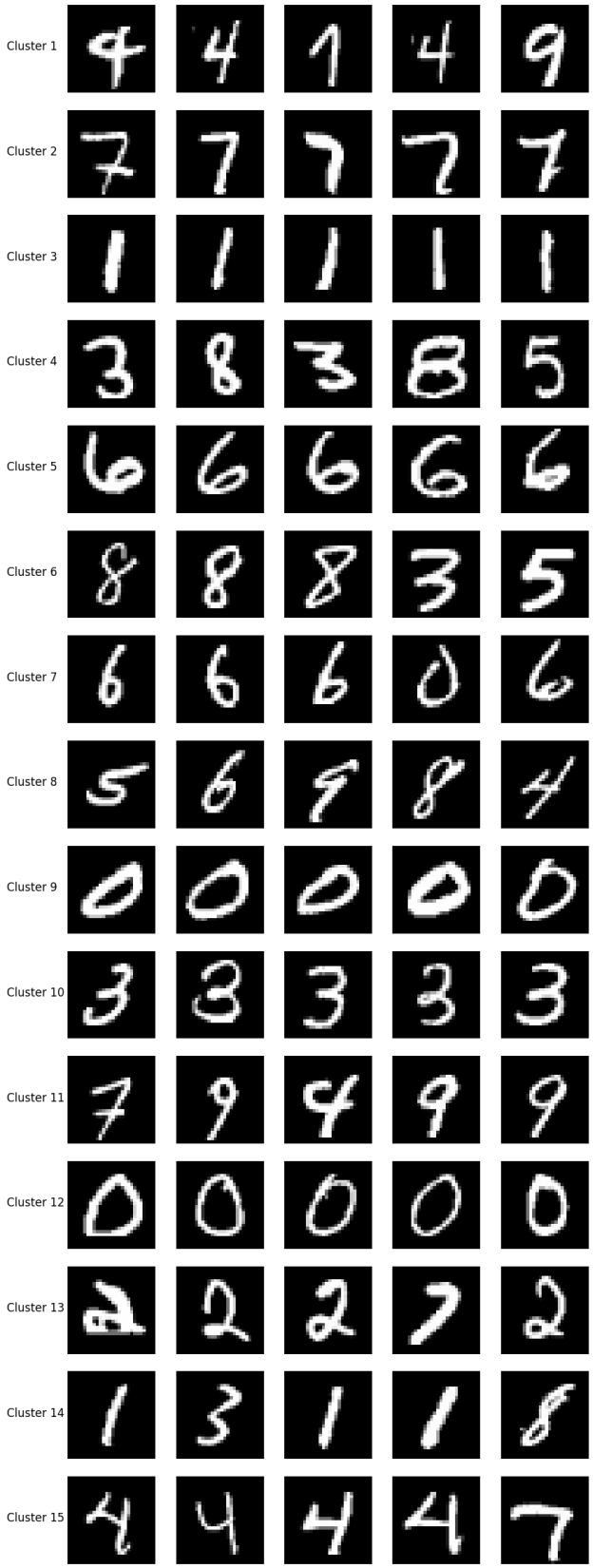

Figure 1: Plot of 5 images from each cluster of the best performing k-means model

From this, we can see that the top k-means model used 15 clusters. As stated above, this is because it split, for example, the digit 1 into multiple clusters (straight 1s vs. slanted 1s) to minimize variance.

Since k-means assumes clusters are circular and of equal variance, it probably failed to capture the single digit 1 as one cluster when its number of clusters was limited to 10 only, and so 15 clusters performed better.

We can also see this in Fig. 1, where there are separate clusters for both types of 1, as well as separate clusters for other numbers that have multiple distinctive shapes.

When made to use 8 clusters, the model's performance dropped, which shows that it was merging digits into single circular clusters because it lacked the flexibility to separate them.

The top gaussian mixture model used 10 components (or clusters), and 'full' covariance. Full covariance lets each cluster have its own covariance matrix, and so clusters are modeled as ellipsoids with specific orientations. This flexibility allowed the model to capture the correlations between pixels better than the other covariance types. 'diag' covariance had the worst performance, as it assumes features are uncorrelated, which isn't true with image data: neighbouring pixels are highly correlated. 'tied' covariance also had poor performance, which was likely because it assumes all clusters share the same variance, whereas clearly a narrow 1 and a wide 8 do not.

Unlike k-means, however, the top gaussian mixture model correctly identified that 10 components was optimal. This suggests that the elliptical cluster shape give this model sufficient flexibility to capture the 10 digit classes without needing to split them further.

Overall, while k-means is faster and computationally cheaper, the gaussian mixture model clearly outperformed the k-means model - it both scored higher and correctly identified 10 as the optimal cluster count. This was due to the flexibility of the gaussian mixture model's elliptical clusters, which due to the 'full' covariance type could be of varying sizes and orientations, as compared to k-means' rigid circular clusters.

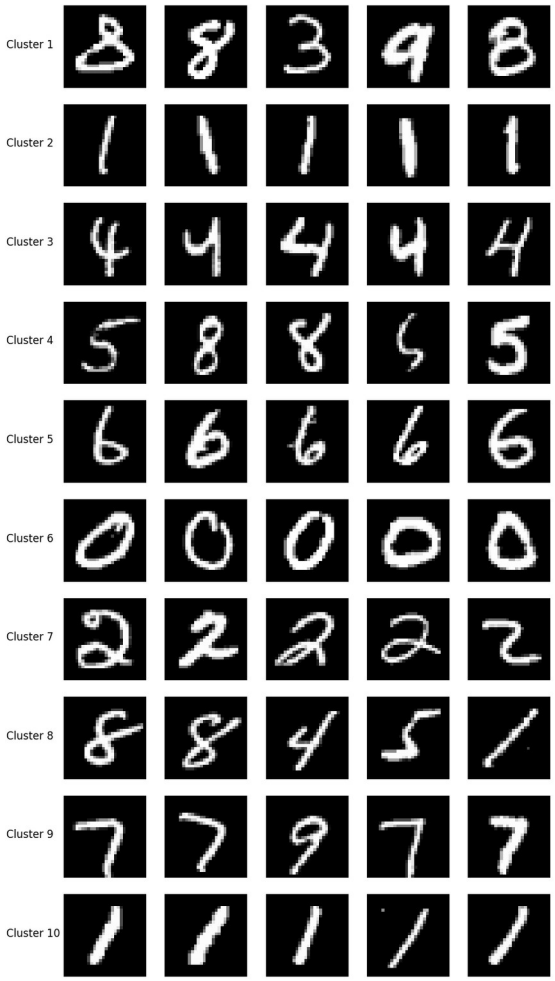

Figure 2: Plot of 5 images from each cluster of the best performing gaussian mixture model