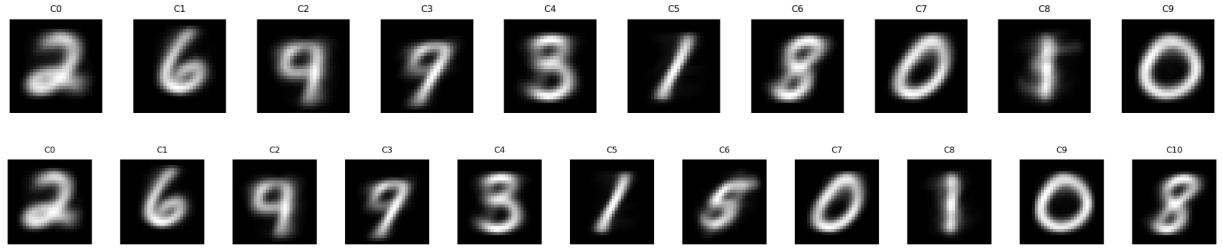
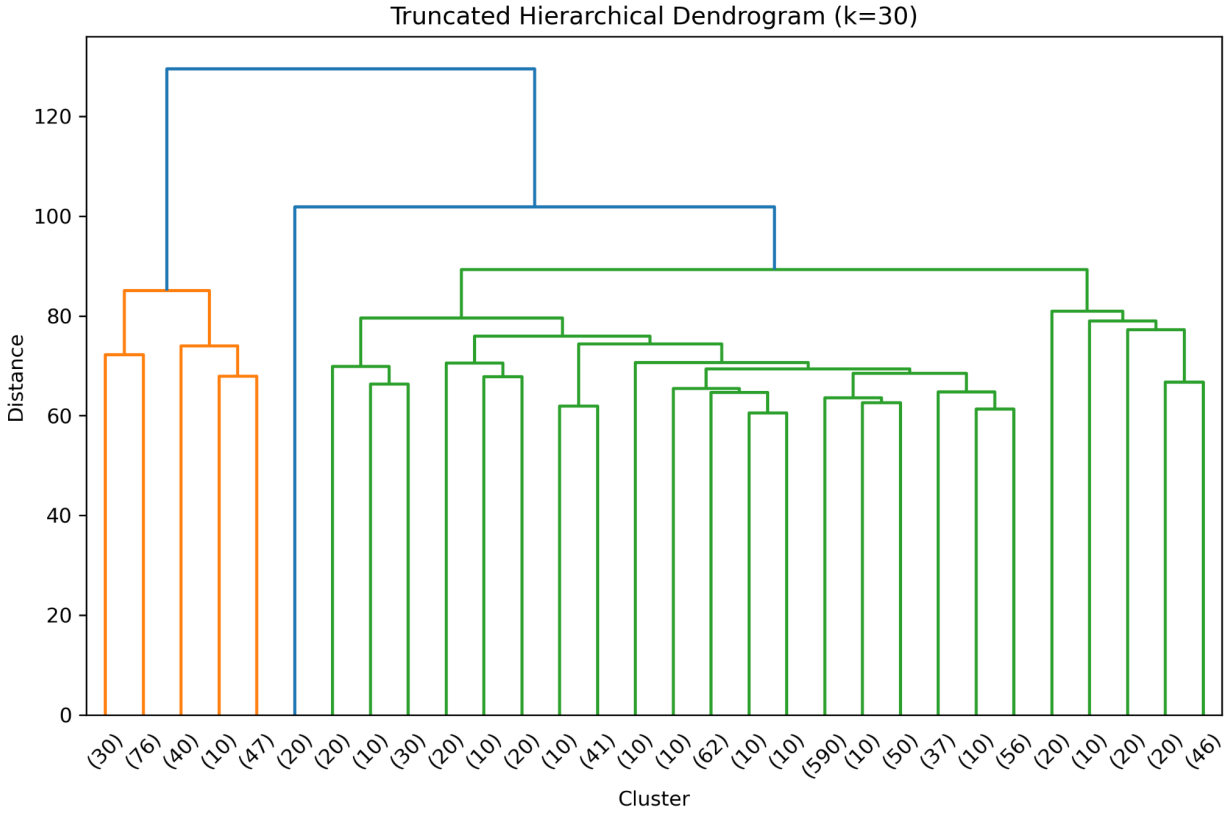
Aaron Morales
BME 205
Fall 2025
2074743

Assignment 3

Images





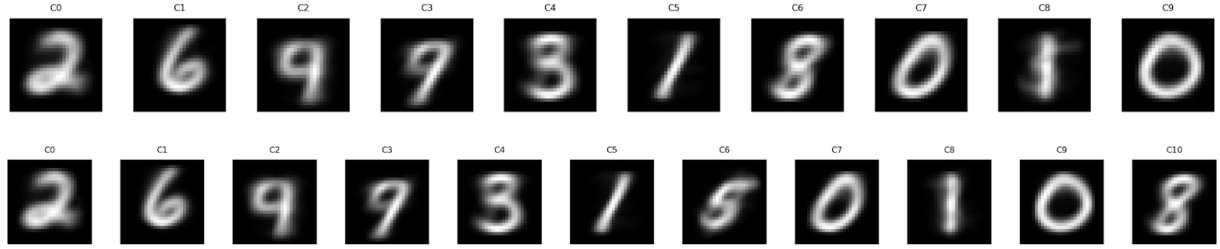
Aaron Morales
BME 205
Fall 2025
2074743

Explanation

Clustering Part One

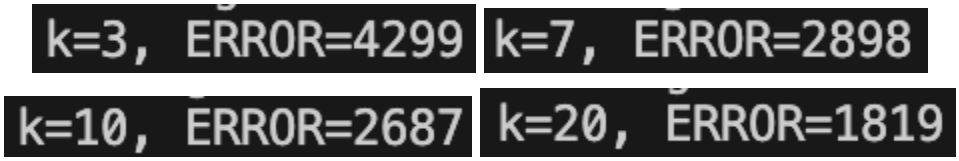
Before exploring the methods to assess the efficacy of our program's results, it's first important to briefly overview the functionality and expected performance of our algorithm. The MNIST dataset provided consists of 6000 hand drawn images(28x28 pixel) of numbers 0-9. Any one of these numbers or samples consists of 784 feature values(corresponding to its pixels 0-255). Also, a provided 1D array holds the labels or true values for each of these samples. The k means clustering algorithm loops the following cycle. K distinct points are chosen at random to serve as the initial cluster centers. In 784 dimensional space, the euclidean distances of each sample are computed and stored. Next, these points/samples are assigned according to the minimum distance centroid. The points are sorted by their new labels and the mean of their feature distributions(in high dimensional space) are computed. These new mean values are then assigned to the new centroid position and the cycle repeats. This loops until the number of max iterations is reached or the centroid no longer moves(tolerance).

To assess the performance of our program there are numerous methods to do so. Firstly, you can analyze the k=10 and k=11 converged centroid images.



I was pleasantly surprised to see coherent numerals at least for the most part. The k=10 run shows obvious 1, 2, 3, 6, 8, and 9 centroids, however it has trouble discerning the differences between numbers like 4s and 9s, 7s and 1s, and 5s and 1s. The pixel values are too similar in order to properly cluster within 10 groups. What's most interesting is that in the k=11 groups, our model is capable of discerning a coherent 5 numeral which was not obvious in the k=10 run.

Visually it seems that increasing the number of clusters(k) may contribute to a smaller margin of error in cluster identity. This can be seen by iterating cluster count…



Aaron Morales
BME 205
Fall 2025
2074743

As k grows as does the number of correctly identified numerals(less room for error) and the k=20 images clearly depict instances of numerals 0-9.

Furthermore, it's important to determine what to attribute the absence of particular numerals in these centroid images. Is it because of pixel ambiguity in the dataset or is it because our algorithm is failing to encompass all of our samples into our k clusters. One sanity check is to print the relative cluster sizes inside within the iterating kmeans loop alongside the final bincount for our predicted labels.

```
Iteration 0: cluster sizes = [ 154  669  123  905 1481 1205  537  233  408  285]
Iteration 5: cluster sizes = [448 519 625 912 839 730 552 318 780 277]
Iteration 10: cluster sizes = [448 528 803 851 769 590 648 337 773 253]
Iteration 15: cluster sizes = [441 541 857 855 746 494 674 350 790 252]
Iteration 20: cluster sizes = [447 535 874 839 732 434 662 350 872 255]
Iteration 25: cluster sizes = [448 532 874 840 730 424 638 350 908 256]
```

```
Converged after 31 iterations (max shift=0.000000)
np.bincount output: [448 533 874 840 729 423 638 351 908 256]
Labels range: min=0, max=9, unique=10
k=10, ERROR=2687
```

The above obviously shows that each numeral is categorized into a bin to some extent. There is high variance between numerals but at no point do they tend to zero even after some 31 iterations!

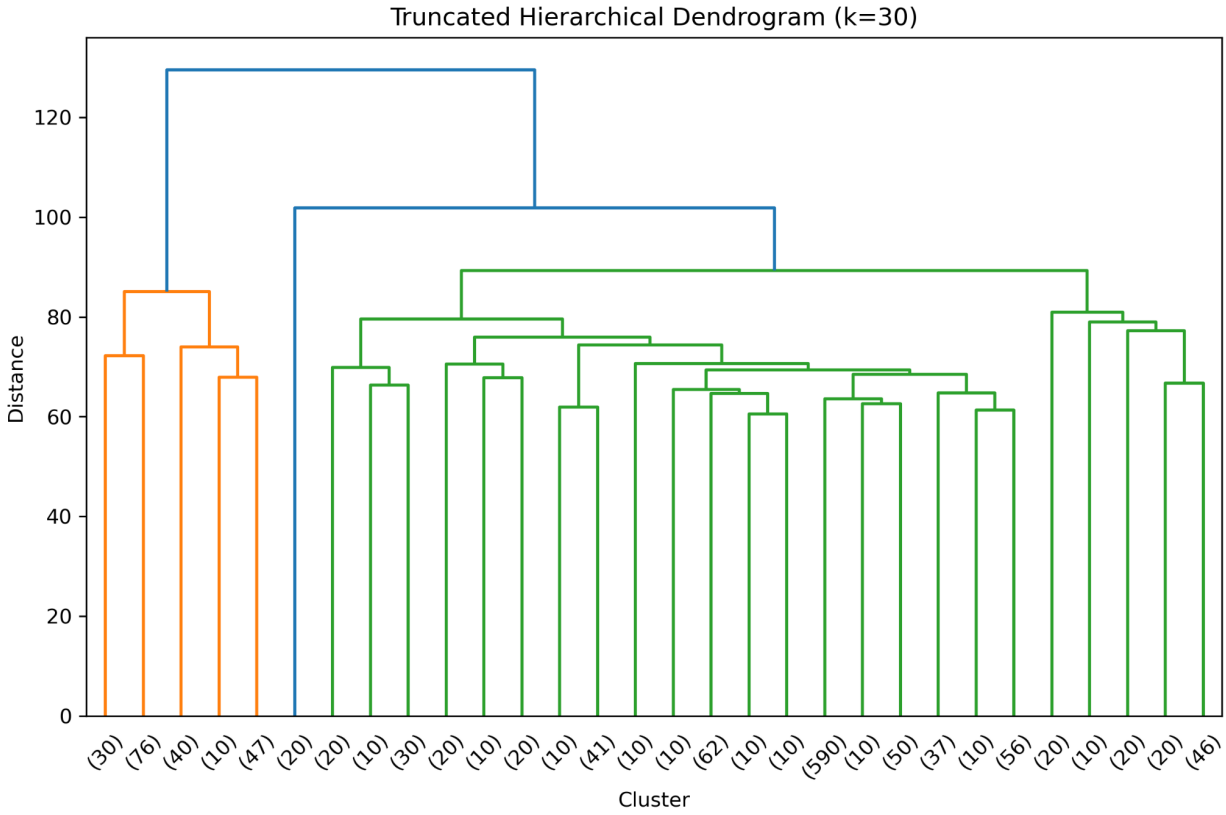
To conclude, with k=10 clusters, the margin of error in classifying our dataset is significant, but through this explanation its apparent that this is no fault of the program but of our dataset. The ambiguity in the sample features makes classification difficult and it would certainly be more effective to implement some form of soft clustering to such a non-distinct feature space.

Clustering Part 2

The program for clustering part 2 is concerned with building a hierarchical clustering of various dog genetic profiles. The SNP feature data provides 1,355 dog samples w/ their respective 178 feature vectors. Alongside this there is a provided labels dataset containing a 1D array of corresponding the true labels for each dog sample. By leveraging ScipPy's Ward linkage method, the program is capable of creating a hierarchical clustering tree in the form of a 30 cluster truncated dendrogram displaying the relative counts of our terminal clusters.

In order to assess the efficacy of the hierarchical clustering program, a great place to begin is to analyze the tree itself.

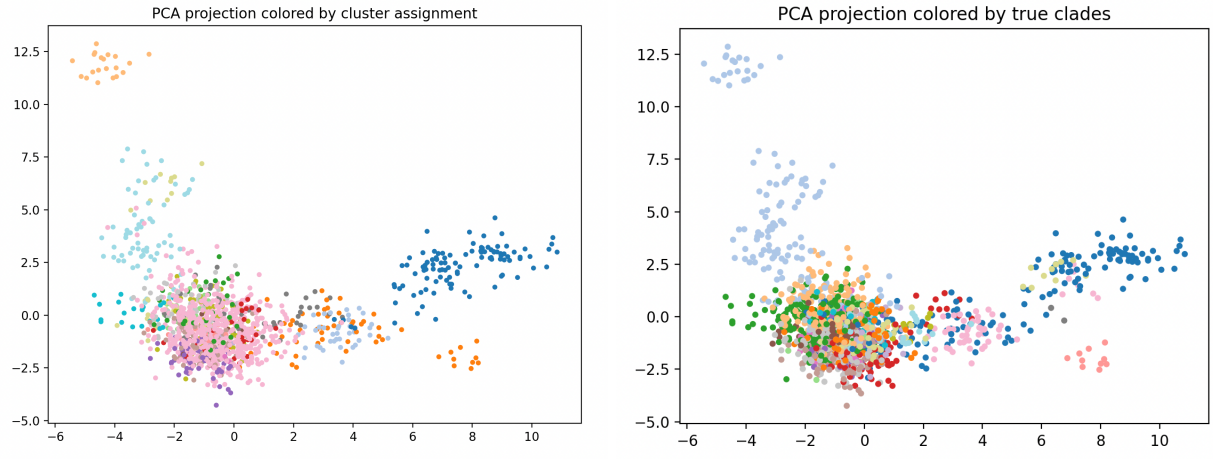
Aaron Morales
BME 205
Fall 2025
2074743



Just from a cursory glance it appears that each cluster has a handful of samples within it, but looking a little closer it becomes obvious that significant portions are classified into one cluster. That 590 count initially seems unusual, but regarding the reference dendrogram of k=n clusters, that branch covers a wide distribution of subsequent clusters not visible to our 30 cluster truncated dendrogram.

Moreover, this may be outside the scope of this assignment, but leveraging PCA dimensionality reduction may shed some more light into orientation of these clusterings and describe the skewed distribution in the truncated dendrogram.

Aaron Morales
BME 205
Fall 2025
2074743



This comparison above demonstrates the discrepancy between cluster assignment counts. The left hand side shows small isolated clusters around the periphery but as the counts begin to conglomerate around the origin, cluster assignments tend to become concentrated in larger and larger clades. As the predicted assignments are engulfed by a single cluster, the true clades show an even distribution of a myriad of clades even around the crowded origin.

To conclude, the margin of error for these 30 clusters sits around 657(mischaracterizations) which is a significant margin for our 1355 samples. This error can undeniably be attributed to the non deterministic nature of our SNP feature space and this is well reflected in our truncated dendrogram as well as our PCA clusters.