Digit Recognition using MNIST Dataset

By: Alex Miller, Kenan Stredic, Shriram Rajasekar

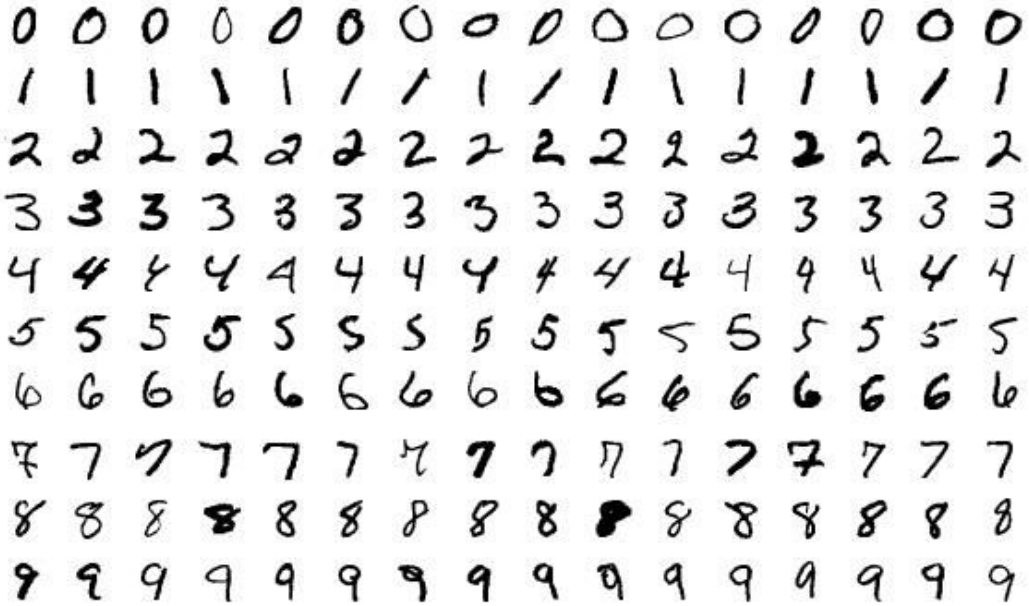
Project Overview

Using the MNIST dataset from Kaggle, which consists of 70,000 images of handwritten digits

The goal is to use a model to determine the digit that each image is representing

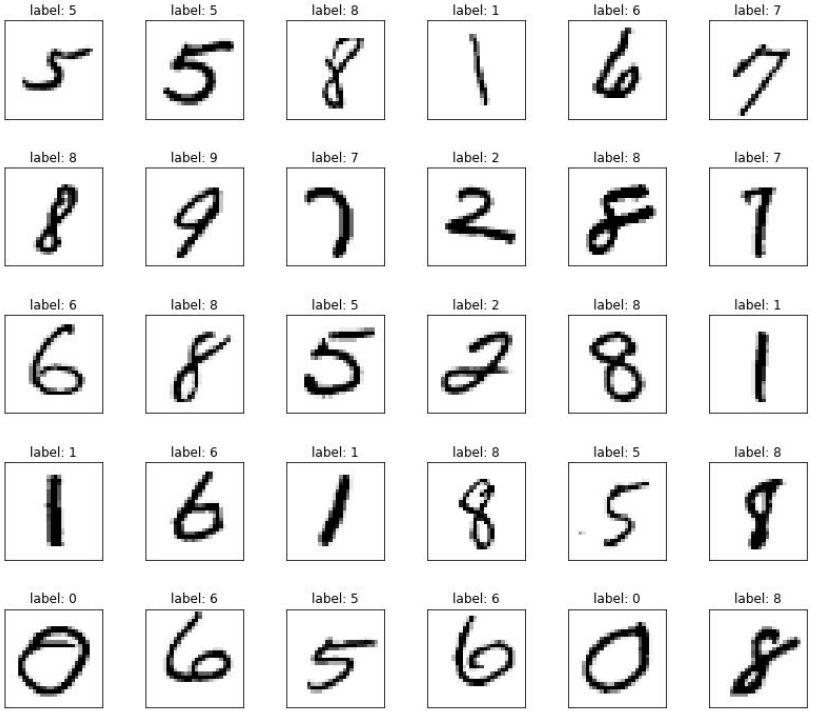
3 methods:

- Gaussian Mixture Model clustering using the EM Algorithm
- K-means Clustering
- Random Forest Classification



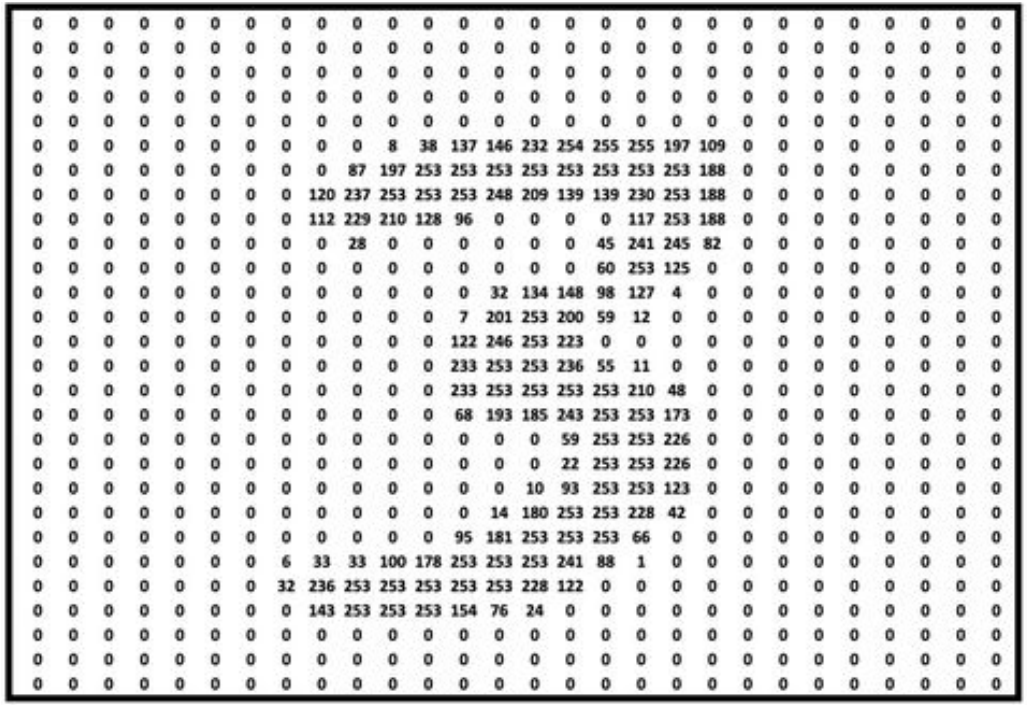
Introduction to MNIST Dataset

- Each image has a corresponding label (0-9)
- Each image is 28x28 pixels - 784 total pixels
- In our dataset, the images have already been converted into numerical data
- 784 pixels per image = 784 variables to represent each image
- Each pixel/variable is given a value of 0 to 255 based on the pixel “strength”, or darkness of the pixel
    - White pixels get a value of 0



Data Preprocessing and Cleaning

- Pixel values were normalized to be between 0 and 1
- Each image was its own matrix, so we flattened the matrices into vectors and row-binded them to form one dataset with all 70k images
- Removed variables/pixels that had a value of 0 for every image since these would have no impact on classifying an image



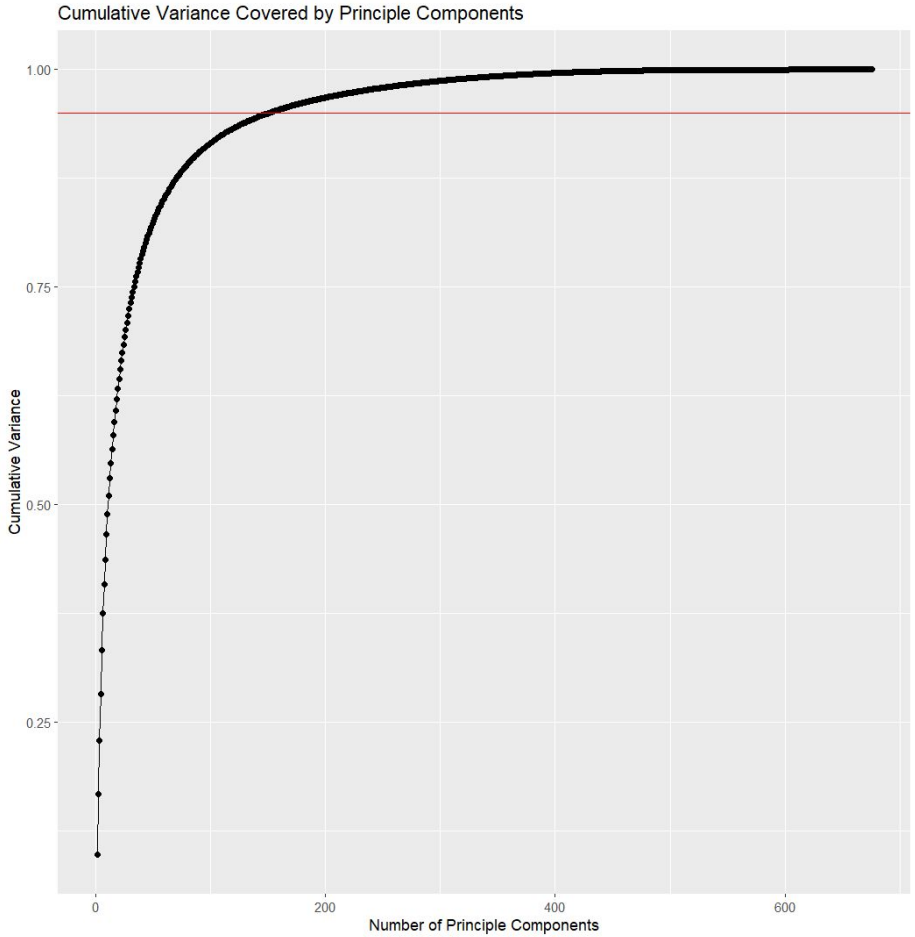
Principal Component Analysis

Dimension reduction is very useful for our dataset

- Can reduce the number of variables from 784 to 150 while retaining 95% of the variance of the data

- Improves model efficiency

The last 500 principle components account for very little of the total variance



Gaussian Mixture Model and EM Algorithm

Used GMM for clustering, where ideally there would be 10 clusters - one for each label

EM algorithm estimates cluster means and variances:

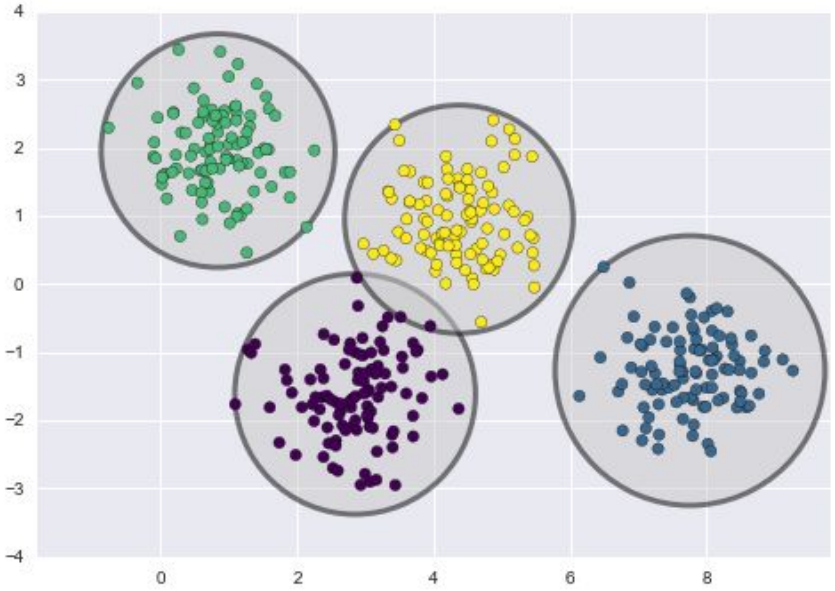
- **GMM is initialized using the labeled data**
- **The initial cluster means and variances are estimated using the means and variances of the labeled data**

E-Step:

- **GMM finds the probabilities an image belongs to each cluster for all images (training & test)**

M-Step:

- **After this, the cluster means and variances are updated using the probabilities found in the E-step**



Note: This is not our data, it is simply for visualization and understanding

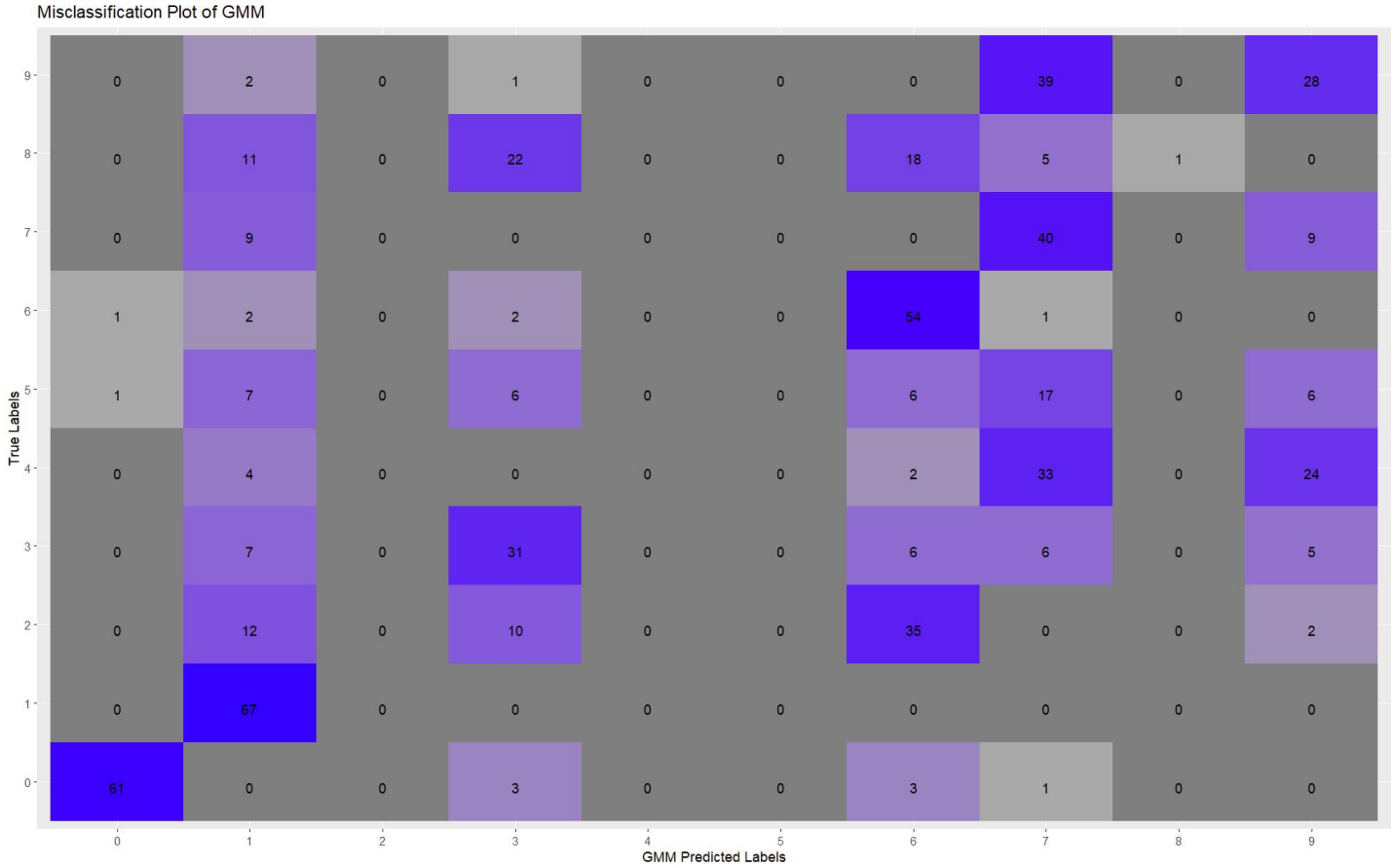
Gaussian Mixture Model Results

GMM method did not perform well

- 47% accuracy

- No clusters for digits 2,4,5

- Barely a cluster for 8



K-Means

- Next, we decided to utilize k-means as an alternative clustering method to determine whether the observed accuracy issues stemmed from the Gaussian Mixture Model (GMM) or other factors.
- Initialization: Randomly sets cluster centroids.
- Training: Iteratively assigns data points to nearest centroids.
- Convergence: Updates centroids based on data mean until stable or max iterations reached.
- Mini-Batch Approach:
  - Instead of using the entire dataset in each iteration, it uses random subsets (mini-batches) of the data.
  - This speeds up computation and is memory efficient, making it suitable for large datasets.
  - The aim was to see if this approach could address size issues caused by clustering.

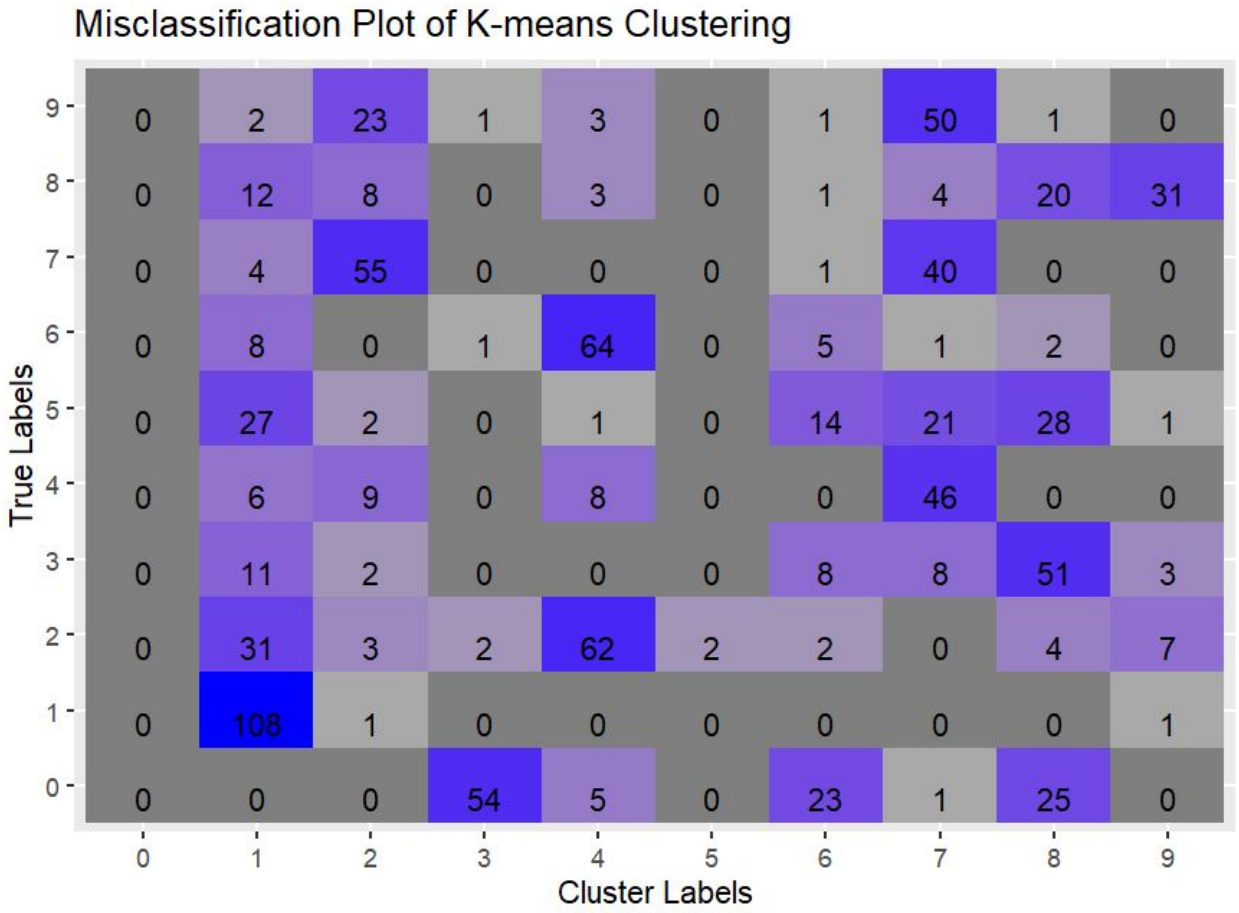
K-Means Results

K-means also struggled with correctly classifying images

- **About 20% accuracy**

Label Discrepancy and Misclassification:

- **The mismatch between true labels and cluster labels suggests the clustering algorithm fails to capture the dataset's structure, leading to misclassification due to misclassified clusters or ambiguous data points.**

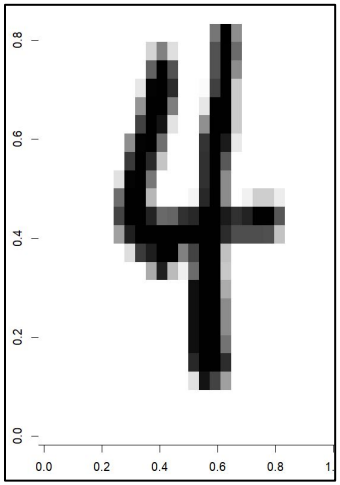


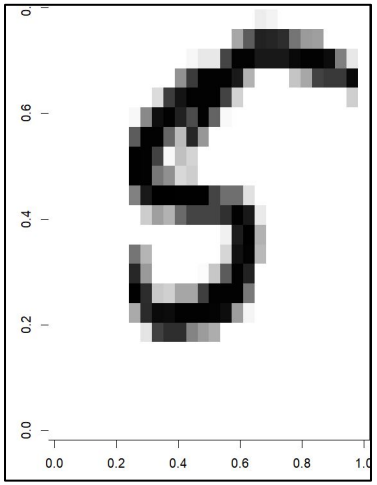
Limitations of Clustering

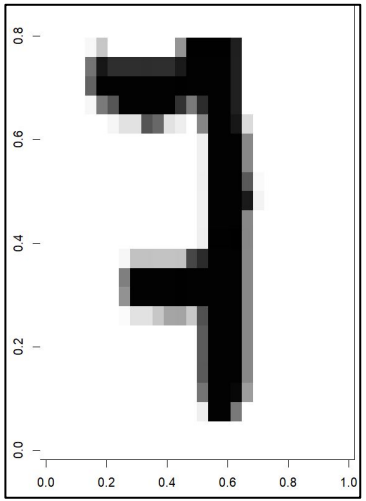
Why is clustering the problem?

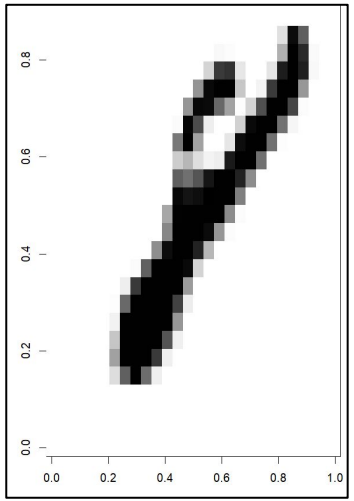
Possible causes:

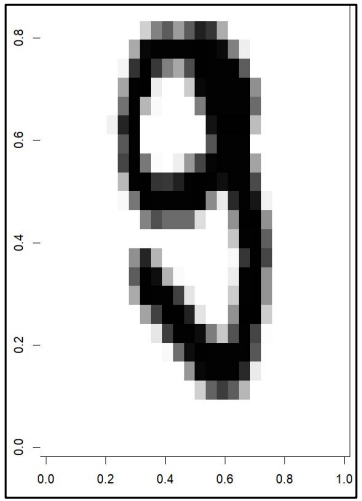
- **Too many variables**
- **Clusters could be too close together**
- **Certain digits have many pixels in common**





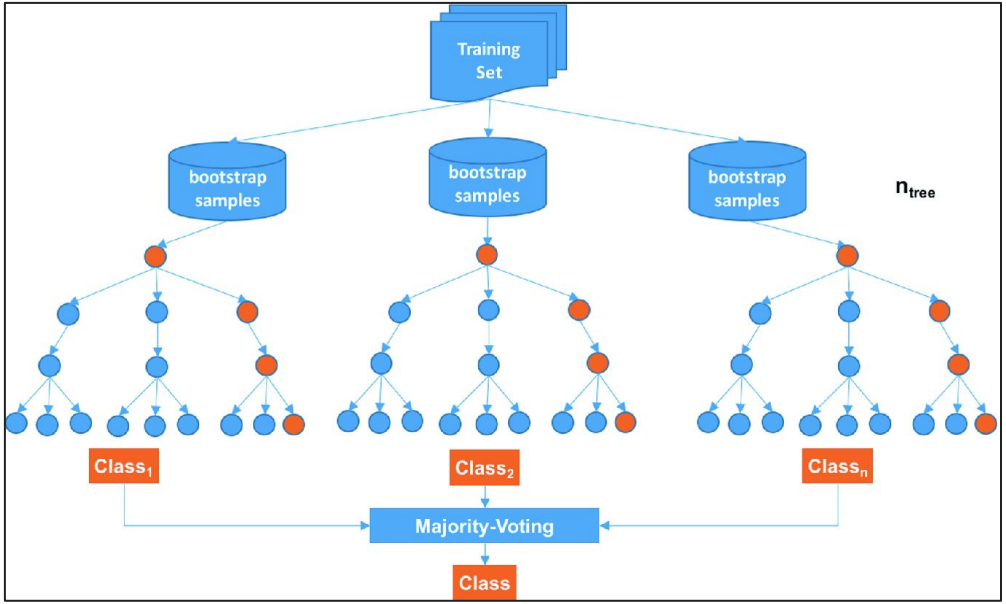






Random Forest Classification

- Creates multiple decision trees using bootstrapping
- Combines the predictions of multiple decision trees to generate the final prediction



Why is it beneficial for our data?

- Scales well to large datasets and high-dimensional feature spaces
- Can handle class imbalances
- Robustness to outliers

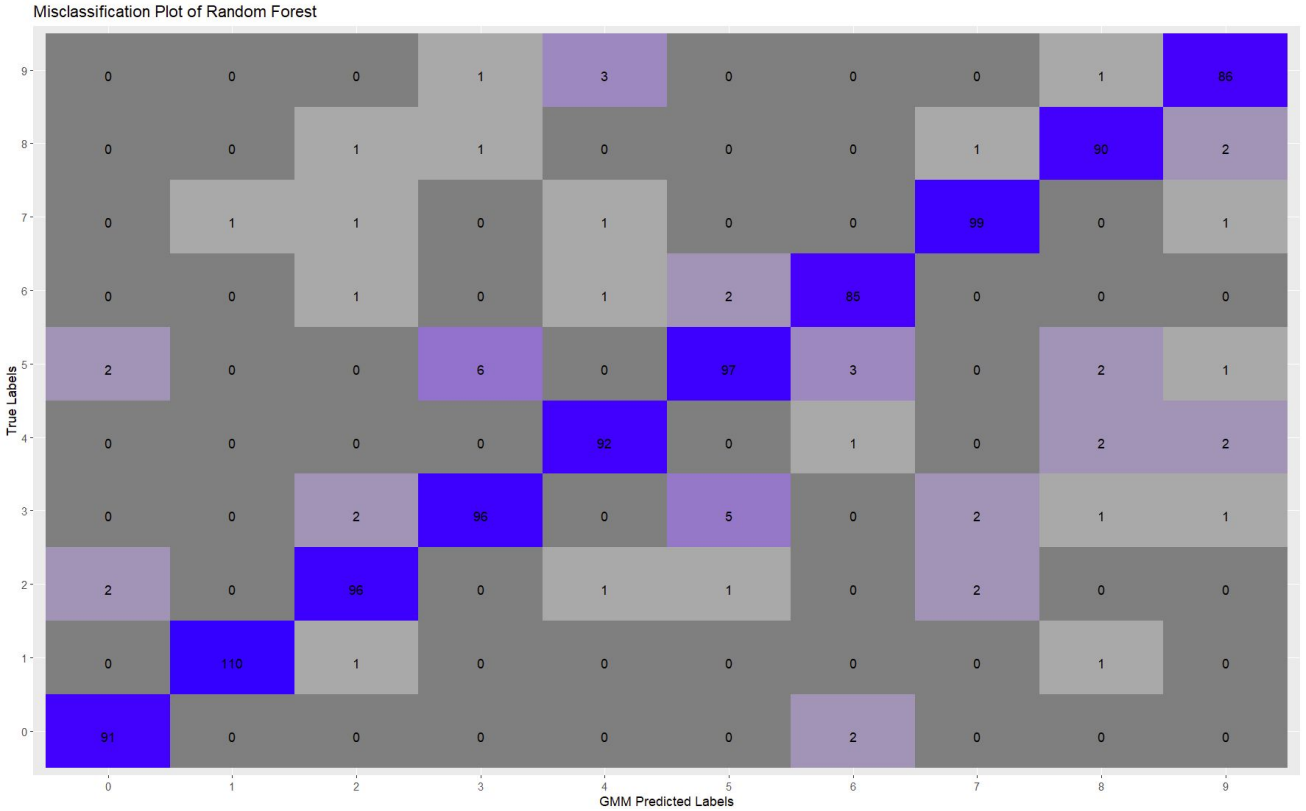
Random Forest Results

This model was more successful at classifying images

- 94% accuracy (with non-PCA data)

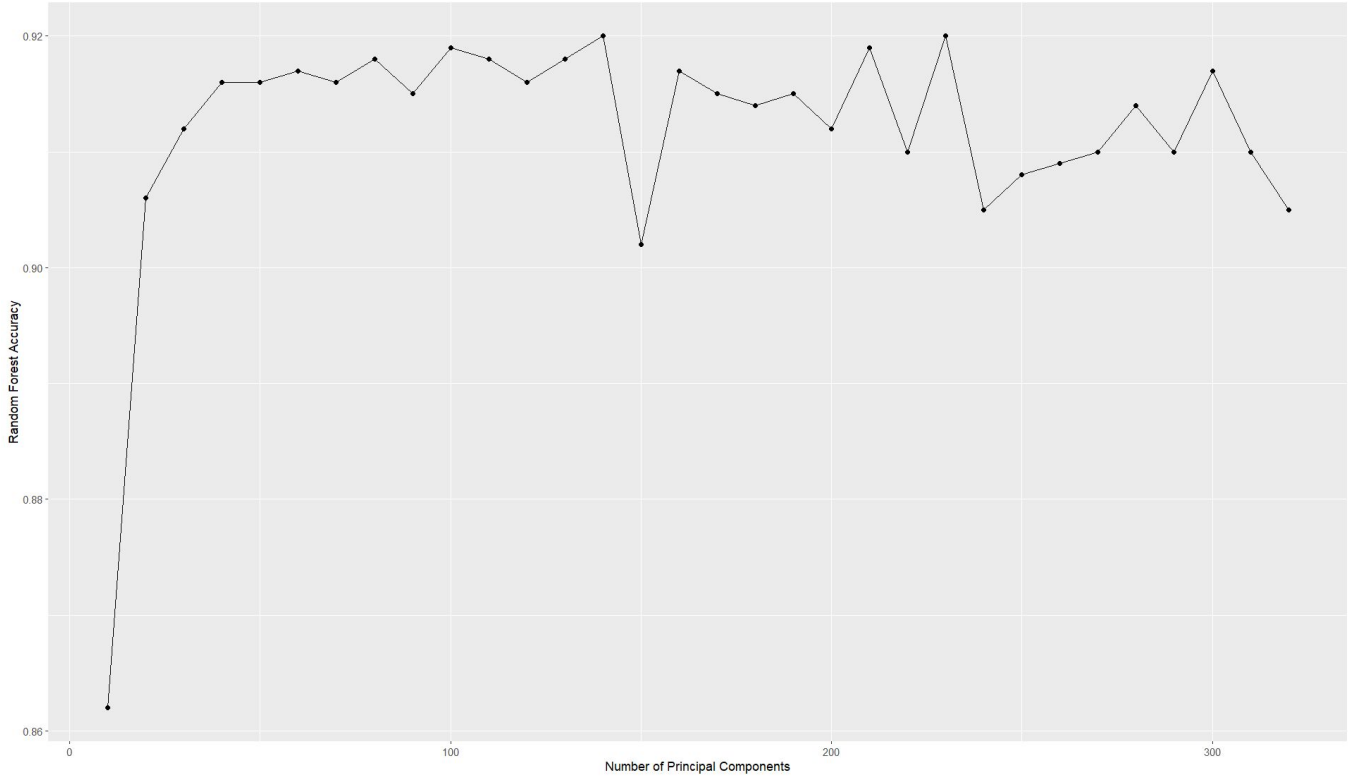
Confusion matrix clearly shows an ability to assign the correct label for most images

Unlike the previous models, random forest shows consistent accuracy for all digits, showing reliability



Random Forest with PCA-Transformed Data

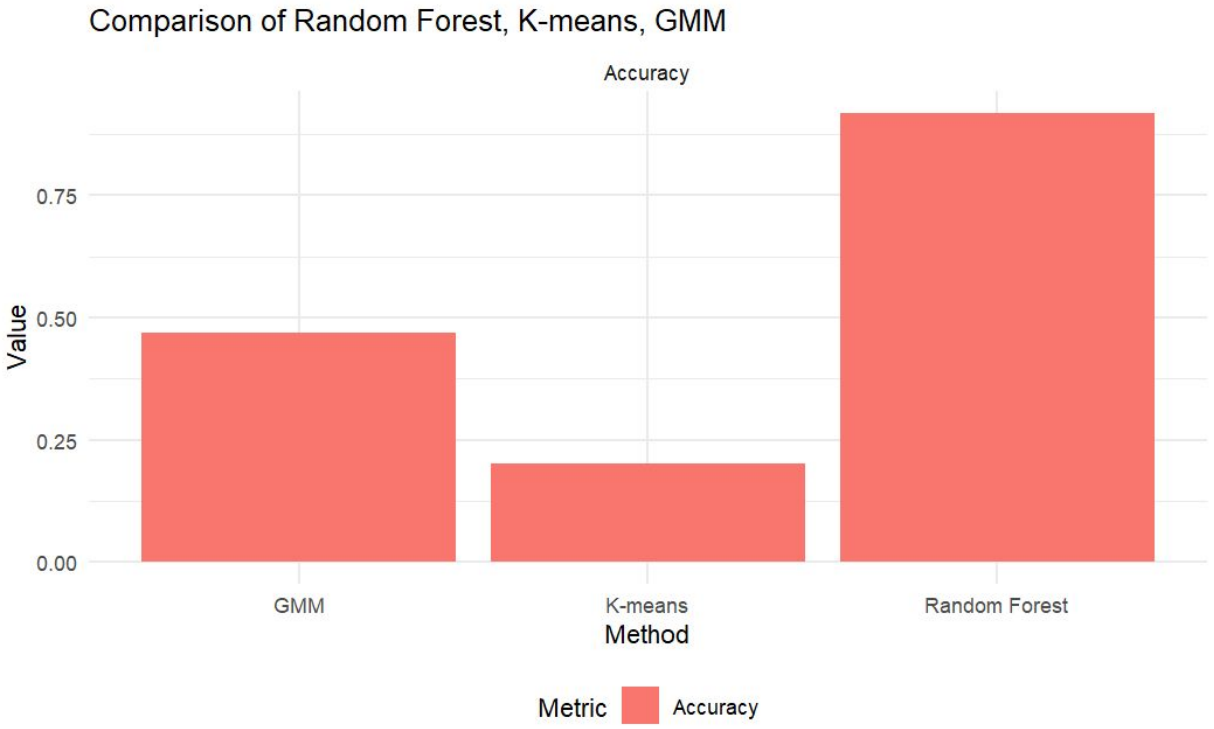
- Overall, PCA was effective at reducing dimensionality while retaining accuracy
- At 40 PCs or greater, the accuracy remains consistent



Comparing Accuracies

- **Random Forest achieved the highest overall accuracy of 94%. However, GMM and K-means exhibited significantly lower accuracies, at 47% and 20% respectively**

- **This disparity can be attributed to the challenges clustering methods face when applied to larger datasets like MNIST without proper data subset selection**



Future Work and Improvements

- Look more into why clustering failed. Is there something we could have done to fix the accuracy issues?
- Ideally, we would like to achieve even higher accuracy for our random forest model
- Try more models to determine if they would work better on our dataset
- Expand this solution to identify letters or multi-digit numbers by classifying each individual digit and concatenating