T2P4_Exploratory_Data_Analysis

March 23, 2025

1 Writing Assignment 3 - Exploratory Data Analysis

Libby Stephan, Leanne Beltman, Seth Bonin, Justin Coffey, Connar Gibbon

1.1 Data Management and Cleaning Strategies

Data Cleaning Because our data is taken from large datasets that have been carefully assembled for various machine learning applications, they have already been thoroughly screened for data errors by their developers prior to release. We have not encountered any issues with missing or improper values while working with our datasets.

For CelebA, the dataset consists of 202,599 facial images, which represents 10,177 unique identities. Each of these images is annotated with 40 binary attributes and 5 landmarks per image (these landmarks indicate where facial features are in the photo). The CelebA images are in color. Examples of these binary attributes include "Wearing Hat" or "Smiling" and values are represented as either -1 (false) or 1 (true) depending on the subject. We do not have to convert any of the annotation data for this project, as it already consists of binary values.

Our Biased MNIST dataset builds on the standard MNIST dataset but provides an additional challenge for image classification by rescaling the target digit, applying a color to the digit, overlay textures, and adding colored letters to the image. The Biased MNIST dataset consists of 70,000 .jpg images and a corresponding JSON file which contains information about the image (such as the target digit, image scale and position, texture, and letter). We did not have to convert any of the annotation or image data for this project.

Data Reshaping Because our CelebA dataset is so large, we frequently ran into computation issues and had to resize images prior to performing dimensionality reduction techniques. Previously, our images were too large to apply the necessary EDA techniques so we resized them to 64x64, which has made them easier to work with. The numerical data has not been adjusted.

Data Management We are following the data management procedures outlined in our second writing assignment, although we have not built a model yet and are simply performing exploratory analysis on the data after it has been downloaded for this assignment.

1.1.1 Methods

In this notebook, we will first examine our CelebA dataset. We will use correlation heatmaps and pairwise frequency calculations to explore which attributes frequently appear together, which may then be difficult for a neural network to distinguish when they appear independently. We will also

examine the distributions of individual attributes, since attributes that are not well represented may be more frequently be misclassified by a model. Finally, we will use the PaCMAP dimensionality reduction technique to determine if any attributes are separable from one another when decomposed. Attributes that are separable may be easier for a neural network to learn, so we will look for attributes that are not easy to separate. PCA and UMAP could not be performed due to computation power restrictions.

Then, we will examine our Biased MNIST dataset using similar techniques. The biased MNIST dataset contains many varieties of the images with varying degrees of correlation between the variables within them. We will look at the datasets with correlations of 0.1, 0.5, and 0.99. We will observe distributions of individual variables, as well as using stacked bar charts to examine multivariate patterns in the datasets. Then, we will use PCA, UMAP, and PaCMAP to perform dimensionality reduction again and look for separable variables.

Our objective is to look for variables that may be underrepresented, highly correlated, and/or not easily separable via dimensionality reduction, since these may be difficult for a neural network to correctly classify.

1.1.2 Dimensionality Reduction Data

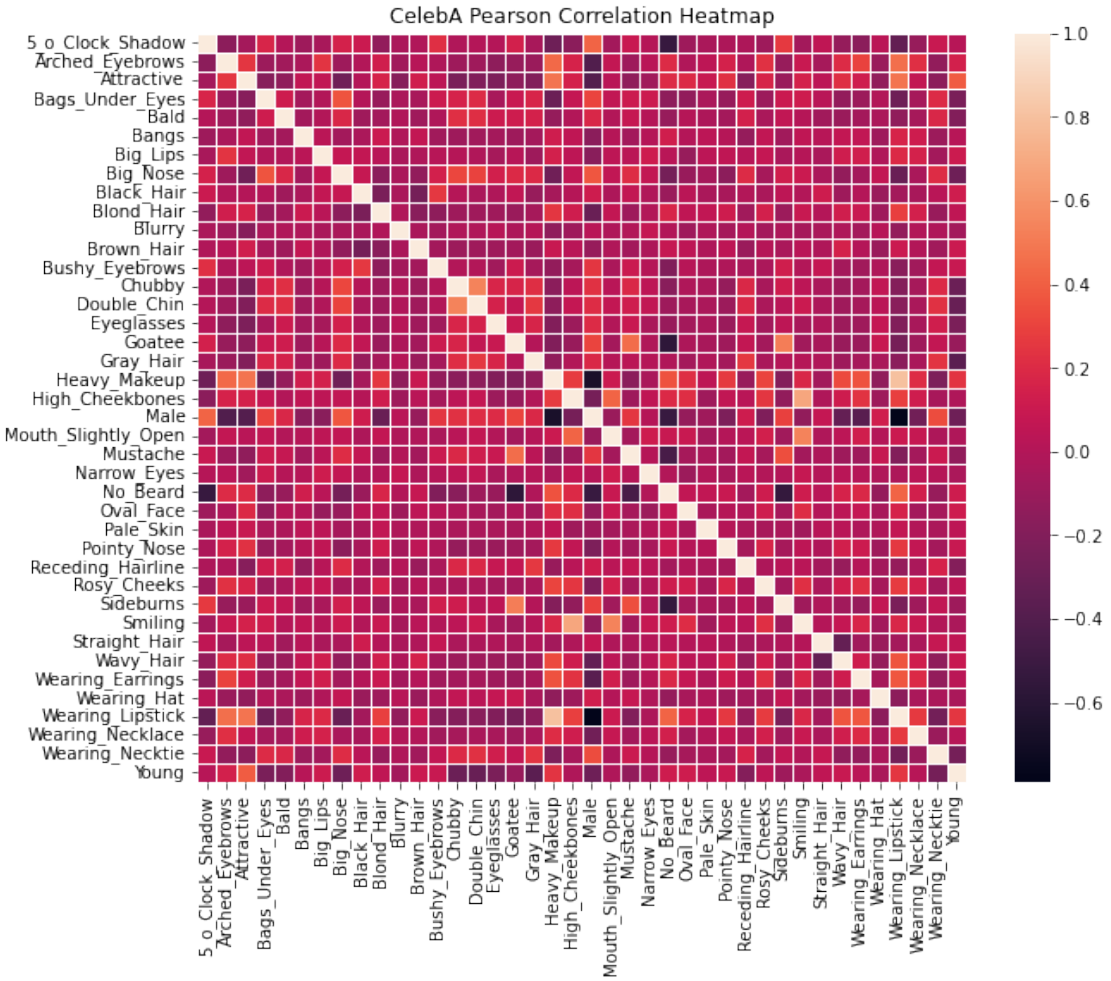
The dimensionality techniques we performed are performed in separate notebooks (found on our Gitlab) and then converted to a CSV file for easier use. We are using the CSV results of dimensionality reduction to produce the plots you see in this notebook. To see and run our dimensionality reduction code, the notebooks are:

1. Capstone_PACMAP_CelebA.ipynb (https://gitlab.eecs.wsu.edu/data_424_2025/t2p4/-/blob/main/Code/EDA/Capstone_PACMAP_CelebA.ipynb?ref_type=heads)
2. Biased_MNIST_PCA_PACMAP_UMAP.ipynb (https://gitlab.eecs.wsu.edu/data_424_2025/t2p4/-/blob/main/Data/MNIST/Biased_MNIST_PCA_PACMAP_UMAP.ipynb?ref_type=heads)

1.2 CelebA Data Exploration

1.2.1 Pearson Correlation Heatmap and Strong Correlations

```
Text(0.5, 1.0, 'CelebA Pearson Correlation Heatmap')
```

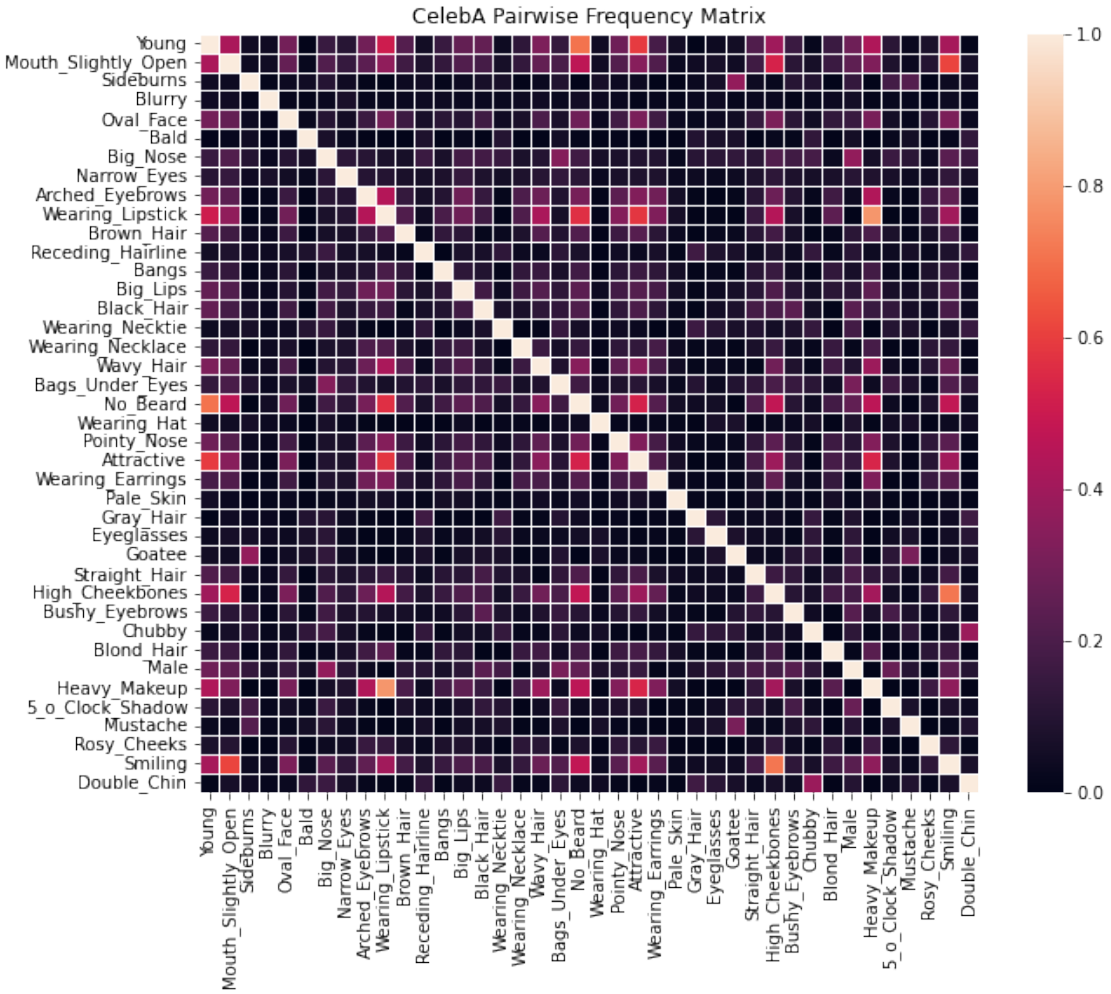



```
Top Positive Correlations Using Pearson
Heavy_Makeup          Wearing_Lipstick    0.801539
High_Cheekbones       Smiling             0.683497
Mouth_Slightly_Open   Smiling             0.536379
Chubby                Double_Chin         0.533713
Goatee                Sideburns           0.512893
dtype: float64
Top Negative Correlations Using Pearson
Male              Wearing_Lipstick   -0.789435
Heavy_Makeup      Male               -0.666724
Goatee            No_Beard           -0.570071
No_Beard          Sideburns          -0.543061
5_o_Clock_Shadow  No_Beard           -0.526946
dtype: float64
```

From the heatmap and obtaining the top positive and negative Pearson correlations among the CelebA attributes, we can get an idea of which variables are most closely correlated with one

another. Features that are highly correlated may be difficult for the neural network to distinguish because they typically appear together. For example, "Heavy Makeup" and "Wearing Lipstick" have a correlation coefficient of 0.80, which indicates a strong positive correlation between them. This means that people that are wearing heavy makeup are also often wearing lipstick. Because they often appear together, it may be difficult for a neural network to distinguish heavy makeup from lipstick. This makes these two variables good candidiates for target variables for our model.

1.2.2 Relative Frequencies

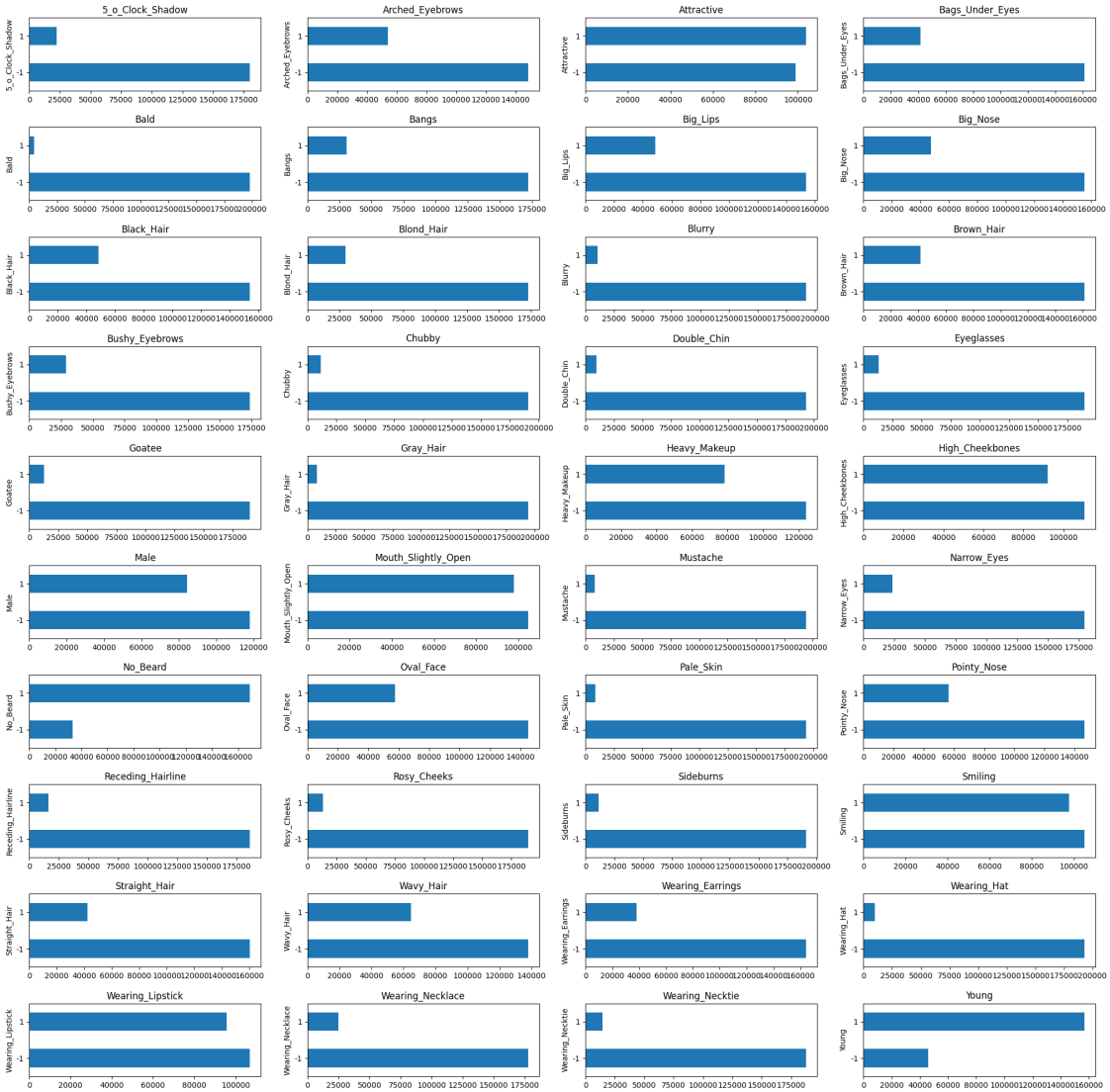



```
Top 5 Highest Frequency Pairs:
('Heavy_Makeup', 'Wearing_Lipstick'): 0.7841
('High_Cheekbones', 'Smiling'): 0.7111
('No_Beard', 'Young'): 0.7036
('Mouth_Slightly_Open', 'Smiling'): 0.6132
('Attractive', 'Young'): 0.5907
```

Again, we can see that "Heavy Makeup" and "Wearing Lipstick" often appear together. This time,

we used a different metric (the intersection of two variables divided by their union, which gives a frequency). We can see that heavy makeup and lipstick frequently appear in photos together, which is in agreement with our findings using the Pearson correlation metric. This further boosts the idea that these two variables may be difficult for a neural network to distinguish because they are often seen together. "High Cheekbones" and "Smiling" also appear in the top 5 for both Pearson and relative frequency metrics, indicating that they could also be difficult to distinguish and good target variable candidates.

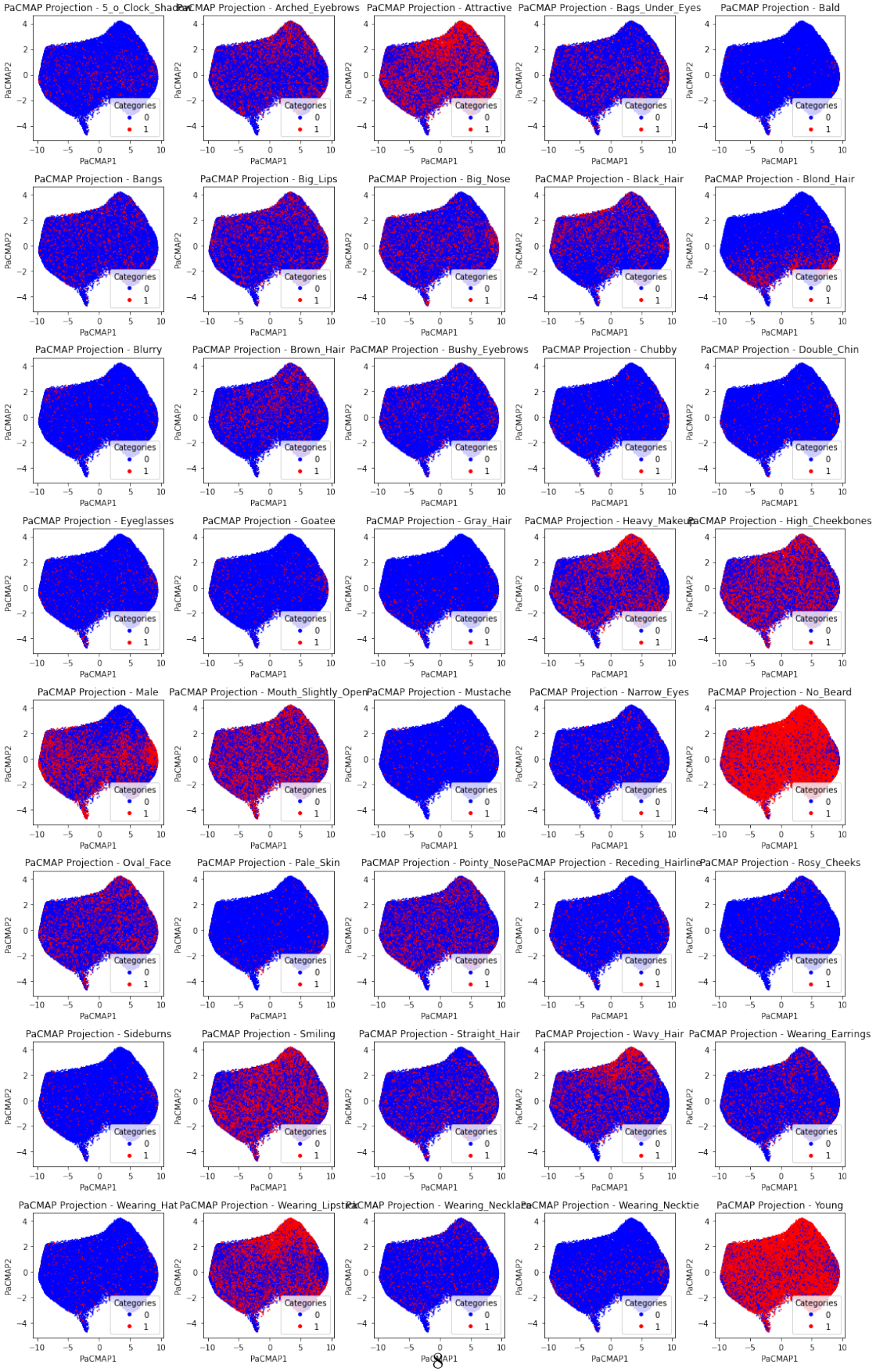
1.2.3 Exploring Attribute Frequency



By looking at the distributions of the 40 binary attributes in CelebA, we can see a wide range of situations. Some variables, such as "Smiling," "Attractive," and "Mouth Slightly Open" are relatively evenly represented across false or true assignments. Other variables are drastically unevenly distributed, such as "Bald," "Wearing Hat," and "Gray Hair," and this can be difficult when

training a model. There are very few people in the CelebA dataset that have grey hair or are bald, so it could be difficult for the model to learn what grey hair looks like since it is not exposed to it often. Traits that are underrepresented in this dataset are good candidates for target variables as well, since it will be difficult for a neural network to understand features it is rarely exposed to.

1.2.4 Dimensionality Reduction - PaCMAP

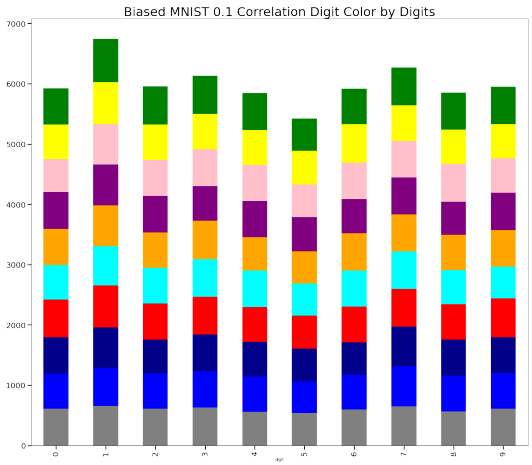


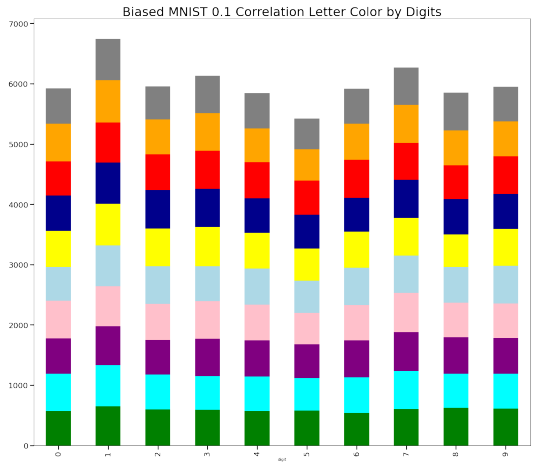
After reducing the dimensionality of our dataset via PaCMAP and plotting each of the 40 CelebA attributes, we can see that most of the traits in the dataset are not easily separable using this dimension reduction technique. If a trait is separable, we would expect to see clear clusters distinguishing the trait. For example, when looking at the plot for "Young" there would be a distinct cluster of images labelled 0 (not young), and images labelled 1 (young). However, for virtually all of our variables this is not the case using PaCMAP. While this is a different technique and can't definitively tell us whether a neural network will be able to accurately classify image attributes, it can give us a sense for what variables may be a challenge to correctly identify. In theory, attributes that aren't separable (which is pretty much all of our variables) may be harder for the neural network to classify and thus better for us to use in our project. This visualization does not point to any obvious variables, but rather tells us that most of the traits in this dataset are not easily distinguishable from one another via PaCMAP.

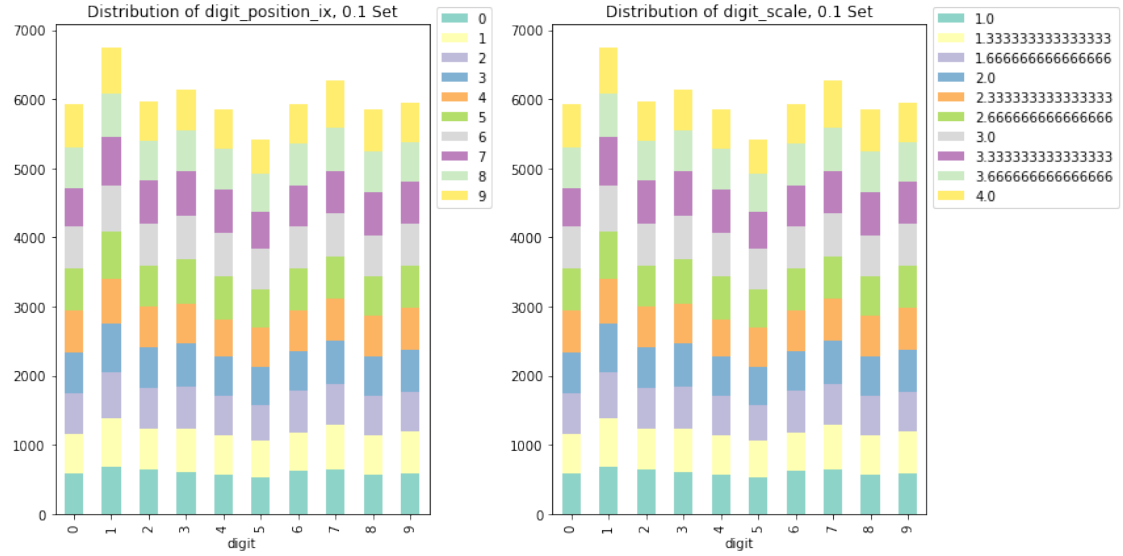
1.3 Biased MNIST Data Exploration

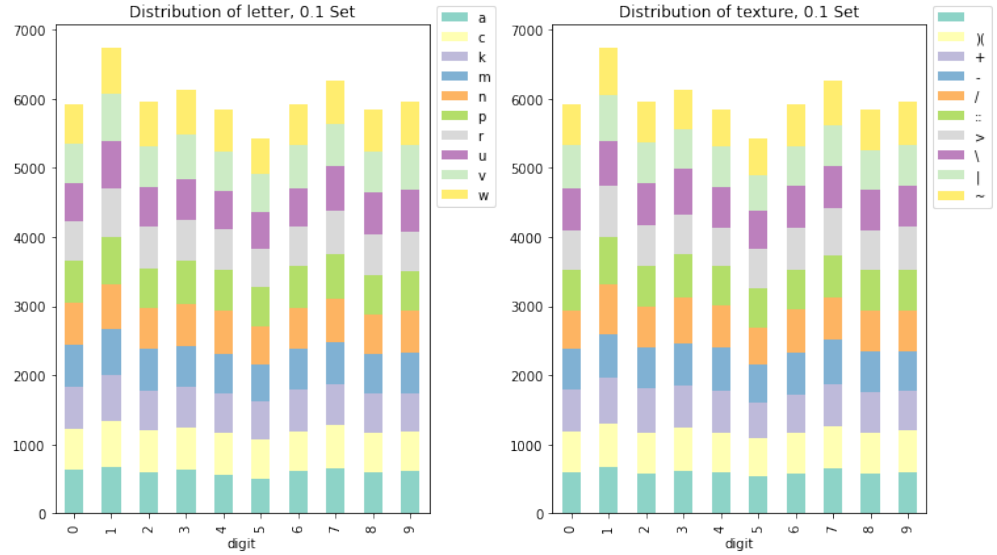
1.3.1 0.1 Correlation Dataset

In the 0.1 correlation set, each variable is approximately evenly distributed across categories. In the sections below, we will use stacked bar charts to identify what these distributions look like when broken down by target variable (digit).


Biased MNIST 0.1 Correlation Digit Color by Digits


Biased MNIST 0.1 Correlation Letter Color by Digits

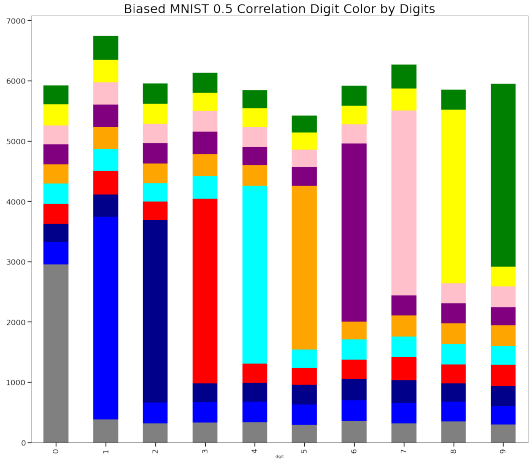

Distribution of digit_position_ix, 0.1 Set
digit
0
1
2
3
4
5
6
7
8
9
Distribution of digit_scale, 0.1 Set
digit
1.0
1.333333333333333
1.666666666666666
2.0
2.333333333333333
2.666666666666666
3.0
3.333333333333333
3.666666666666666
4.0

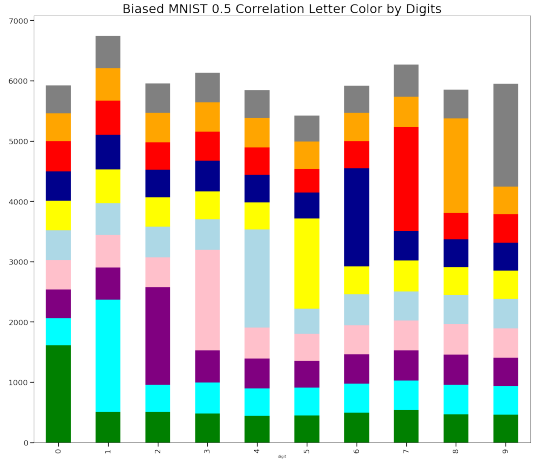

Distribution of letter, 0.1 Set
digit
a
c
k
m
n
p
r
u
v
w
Distribution of texture, 0.1 Set
digit
)(
+
-
/
::
>
\
|
~

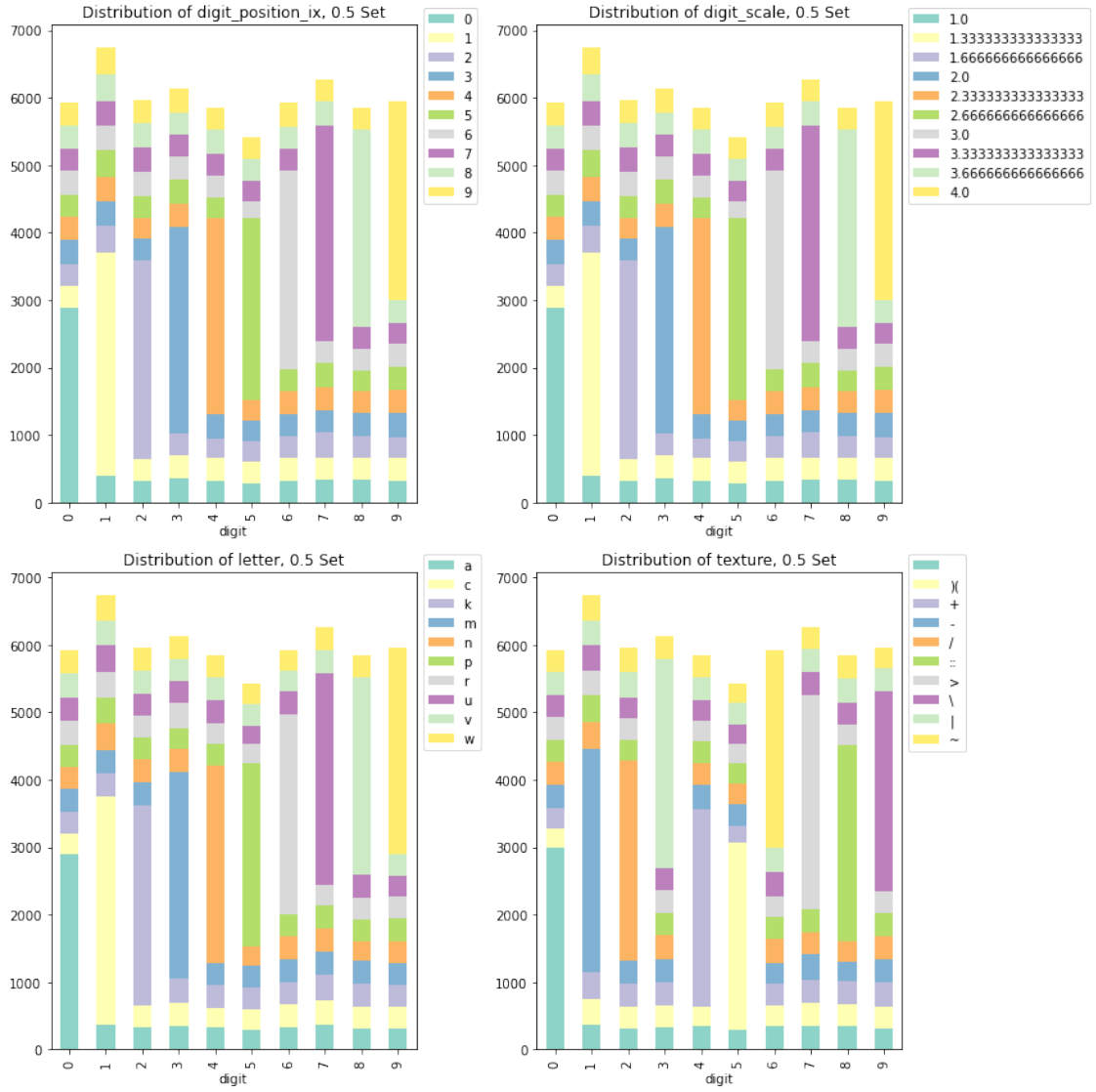
Each digit has relatively equal color distribution for the digit itself and the letter in the image for the 0.1 correlation set, which is not surprising. Because the colors are so evenly represented, it is easier for the neural network to learn them because it is exposed to each color in similar frequencies. The same held for other image properties. Texture, letter, digit position, and digit scale are all approximately evenly distributed across digits. This will make it easier for the neural network to make accurate predictions because it has been exposed to different features at similar rates. This is ideal for training an less biased model, however that is not the goal of our project. We are seeking to create an intentionally biased model to fix, and using a balanced dataset will mitigate bias that could come from class imbalance. Thus, the 0.1 correlation MNIST dataset is likely not a good choice to build our model with.

1.3.2 0.5 Correlation Dataset

In the 0.5 correlation set, each variable is still approximately evenly distributed across categories, although slightly less evenly than in the 0.1 correlation dataset. We will use stacked bar charts to identify what these distributions look like when broken down by target variable (digit) below. We are anticipating that distributions of variables across digits will become more uneven as the correlation of the dataset increases.



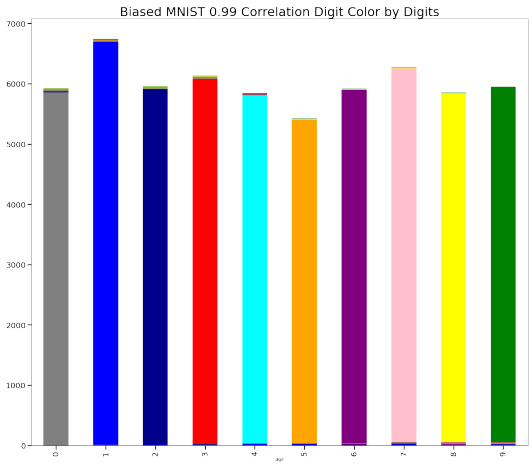


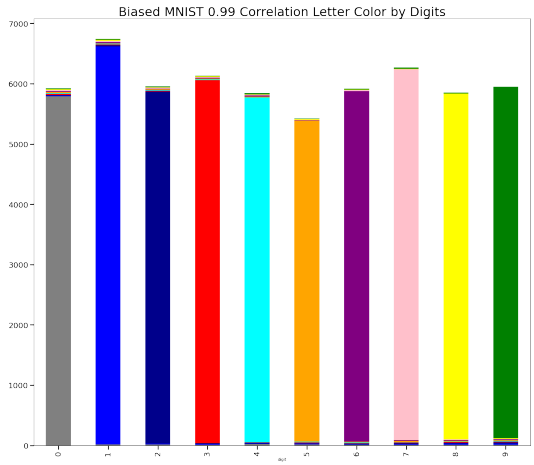


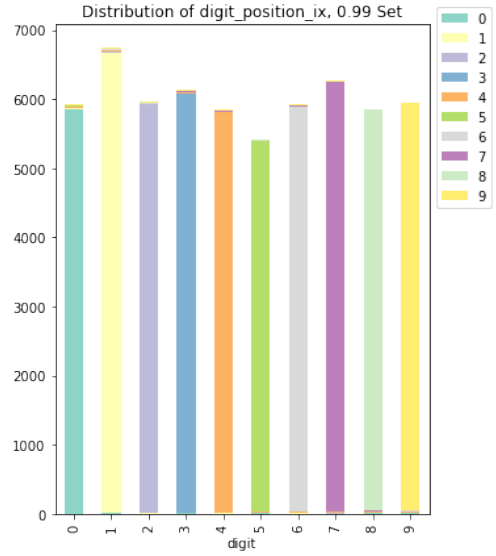
In the 0.5 correlation dataset, we can see that distributions of variables are becoming more uneven across digits. For example, the majority of images that have a 1 as the digit have an "a" as the letter, no texture, an image scale of 1.0, and the digit is at 0. The image color is most often grey, and the letter is green. Other variable categories are represented for each digit, although much less frequently (approximately 300 images per class for the non-dominant class, while the dominant class is present in about 2800 images). This will likely introduce some bias in our model. Since the model has most frequently been exposed to the dominant class, it will have trouble learning about the non-dominant classes with the same accuracy. Similar to human learning, neural networks have difficulty understanding attributes that they are not frequently exposed to, which makes this dataset a better candidate for use in our project than the 0.1 correlation dataset.

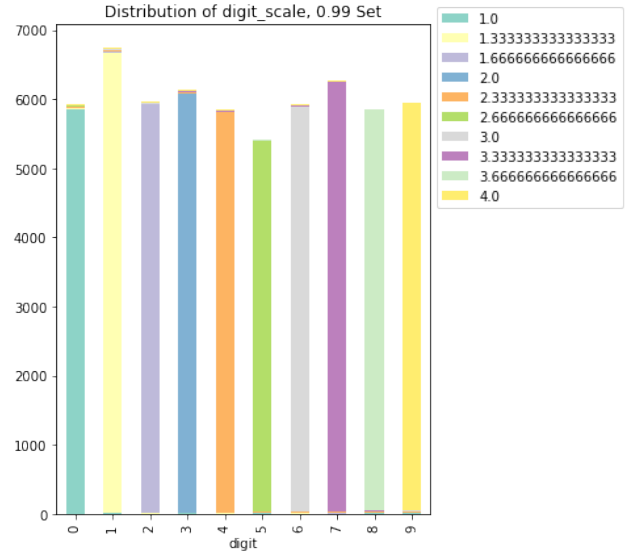
1.3.3 0.99 Correlation Dataset

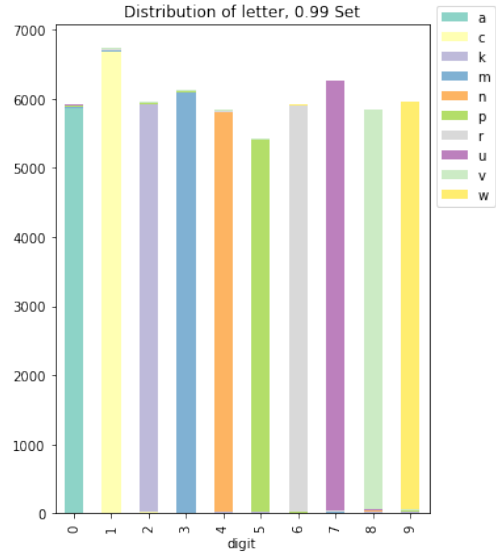
For the 0.99 correlation dataset, each class for the variables are decently well-represented, although not as evenly as in the 0.1 correlation dataset that we examined earlier.

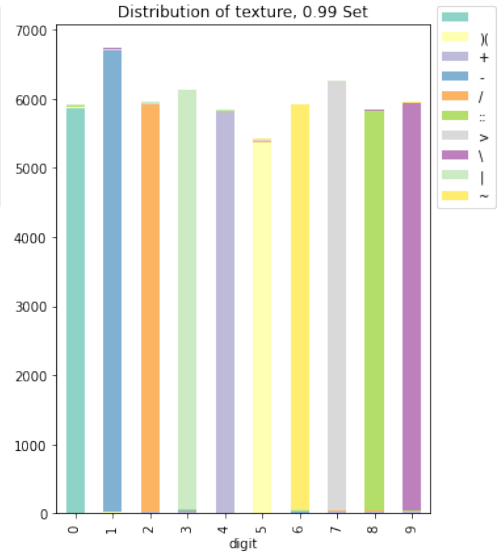

Biased MNIST 0.99 Correlation Digit Color by Digits


Biased MNIST 0.99 Correlation Letter Color by Digits


Distribution of digit_position_ix, 0.99 Set
digit
0
1
2
3
4
5
6
7
8
9

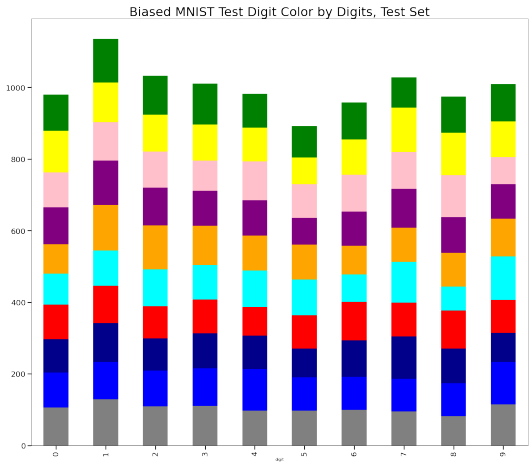

Distribution of digit_scale, 0.99 Set
digit
1.0
1.3333333333333333
1.6666666666666666
2.0
2.3333333333333333
2.6666666666666666
3.0
3.3333333333333333
3.6666666666666666
4.0

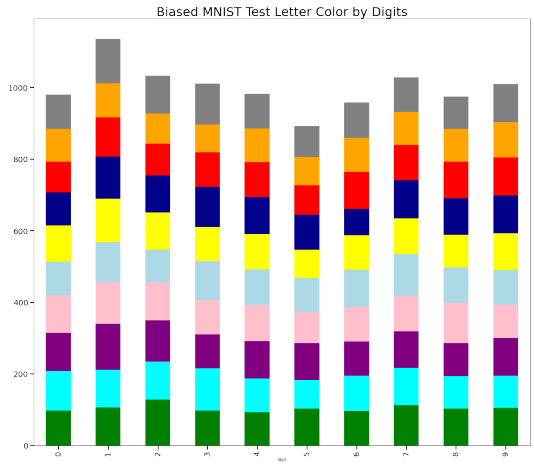

Distribution of letter, 0.99 Set
digit
a
c
k
m
n
p
r
u
v
w

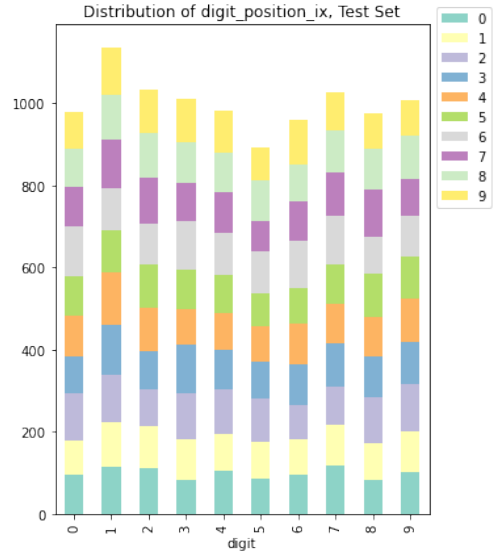

Distribution of texture, 0.99 Set
digit
)(
+
-
/
::
>
\
|
~

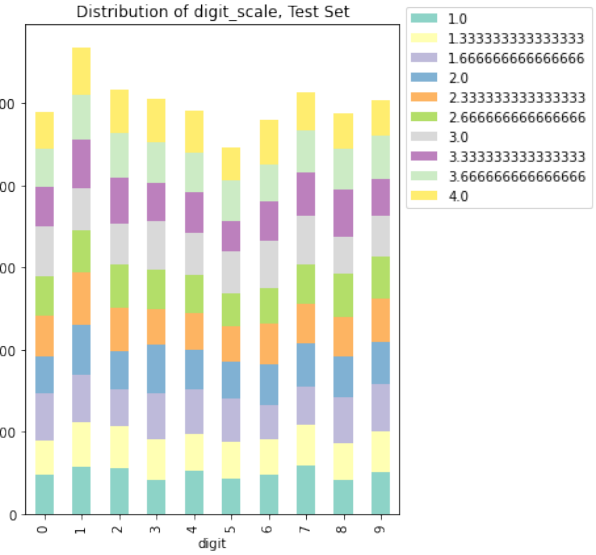
In the 0.99 correlation dataset, we can see extremely uneven distributions of variables across digits. Now, the non-domimant classes account for only ~50 images each, while the domimant class appears in the rest of the images (typically between 5,500-6,500 images for each digit that contain the domimant class). This dataset is likely to introduce the most bias from uneven class distributions into our model since there is virtually no exposure to the non-dominant classes for the model to learn from. The extremely small number of non-dominant class images could make mitigation techniques such as resampling difficult though, since the network would be learning the same 50 images over and over. The 0.99 and 0.5 datasets are our best options for use that we have explored.

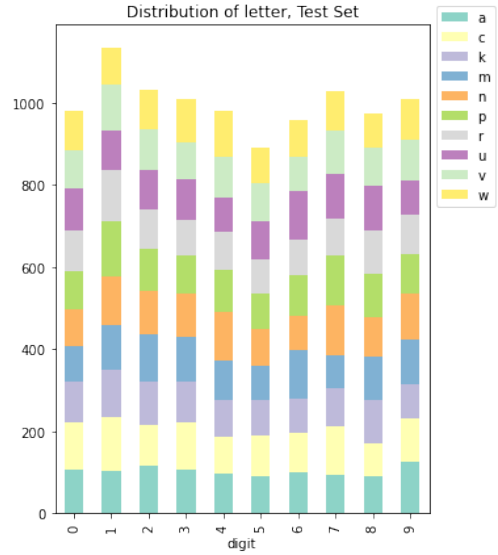
1.3.4 Test Dataset

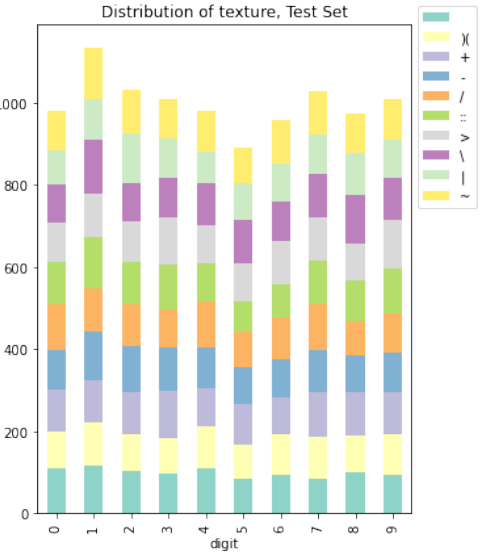

Biased MNIST Test Digit Color by Digits, Test Set


Biased MNIST Test Letter Color by Digits


Distribution of digit_position_ix, Test Set
digit


Distribution of digit_scale, Test Set
digit

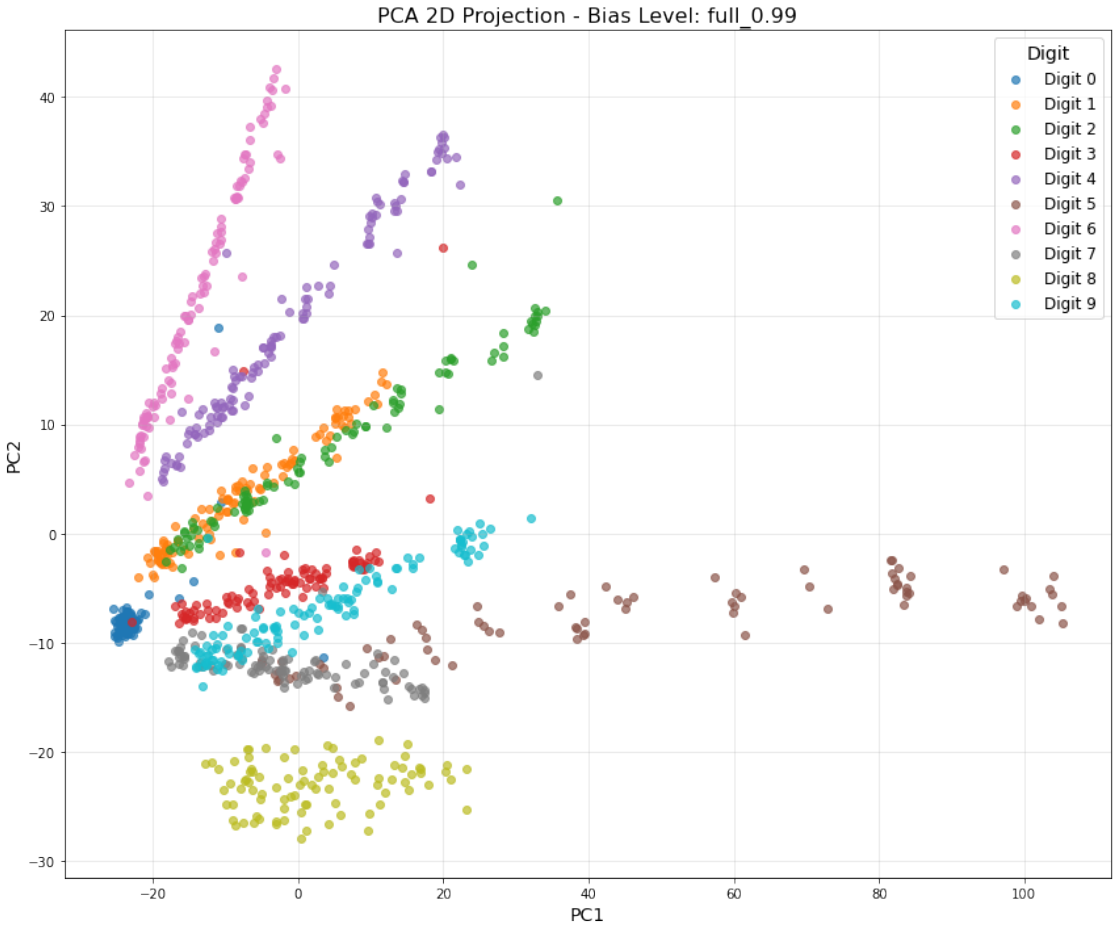

Distribution of letter, Test Set
digit

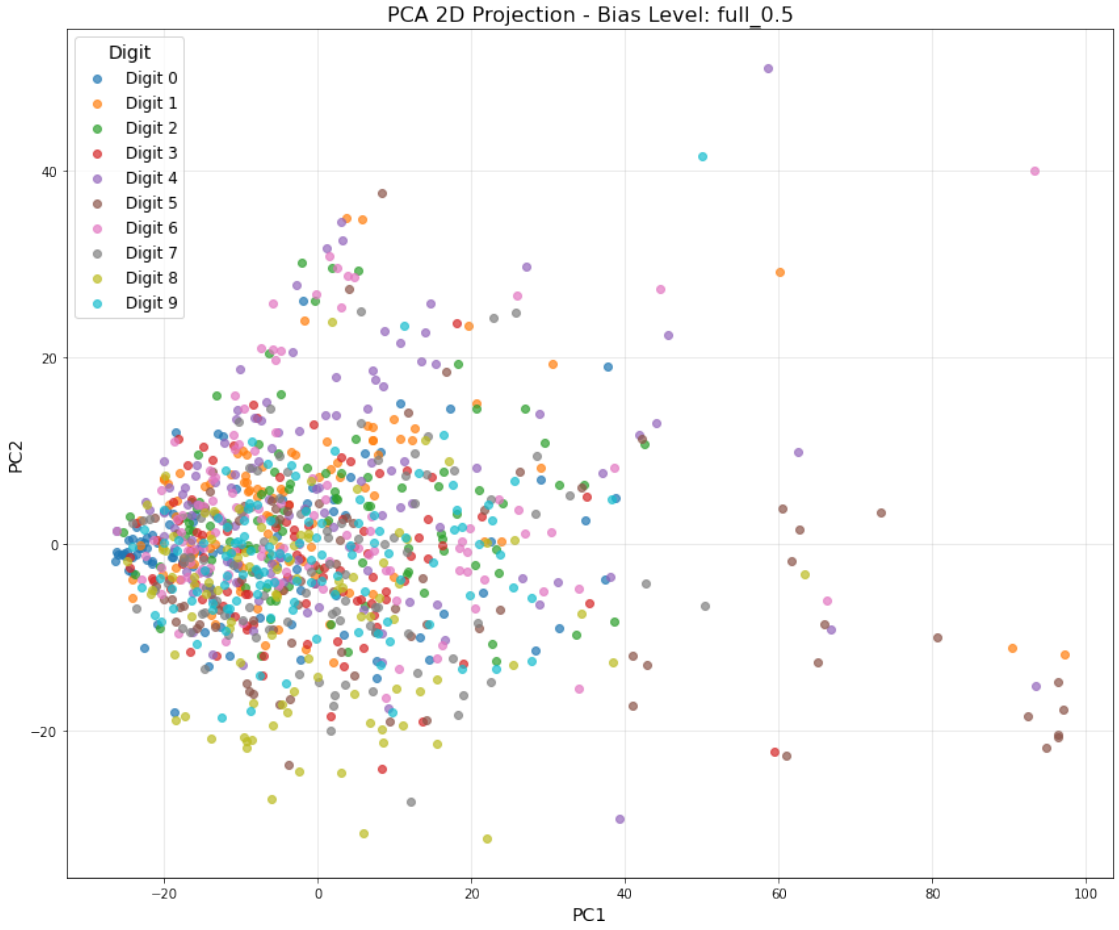

Distribution of texture, Test Set
digit

The test dataset contains very even distributions across all classes, which will be useful to test with. If we train our model on an extremely unbalanced dataset (like the 0.5 and 0.9 correlation datasets), it will struggle to classify a balanced dataset like this one.

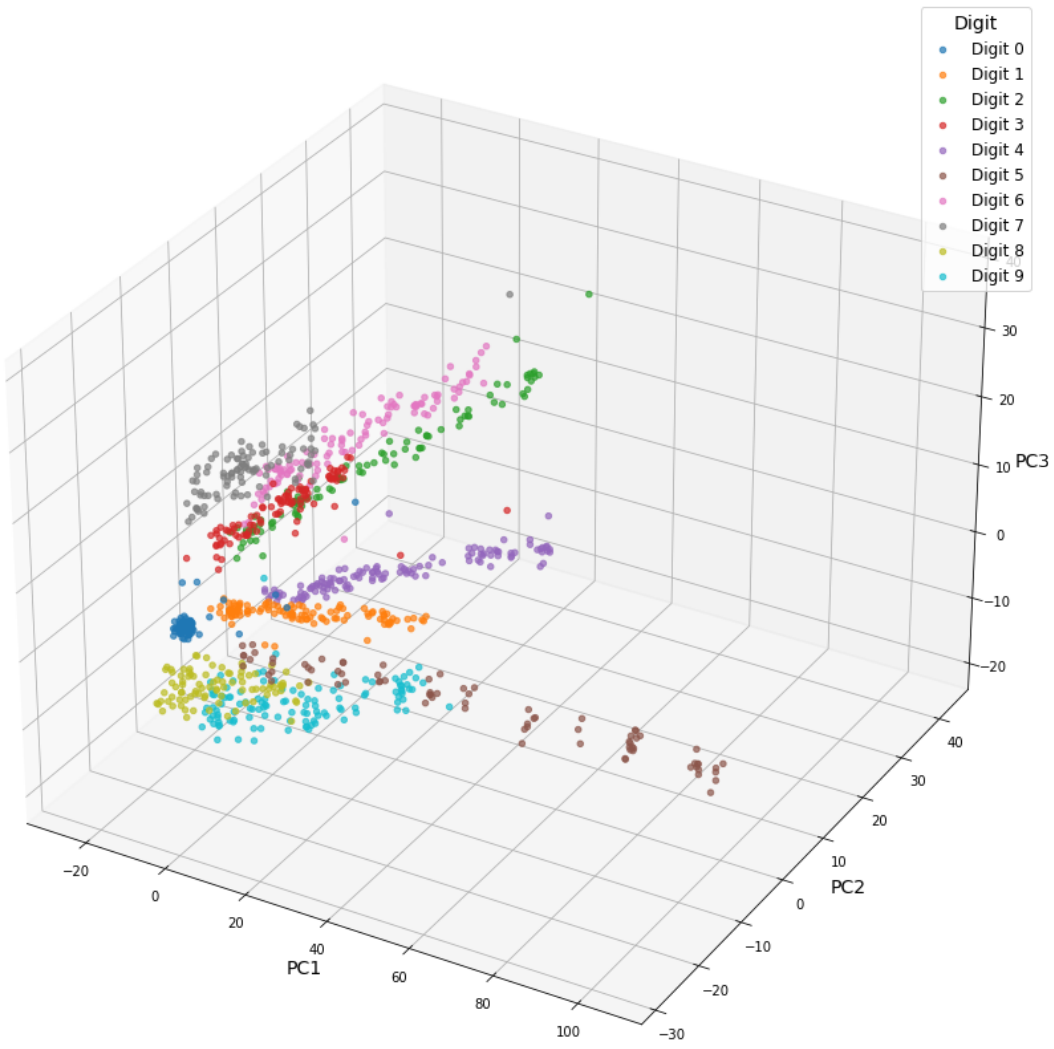
1.3.5 Dimensionality Reduction - PCA

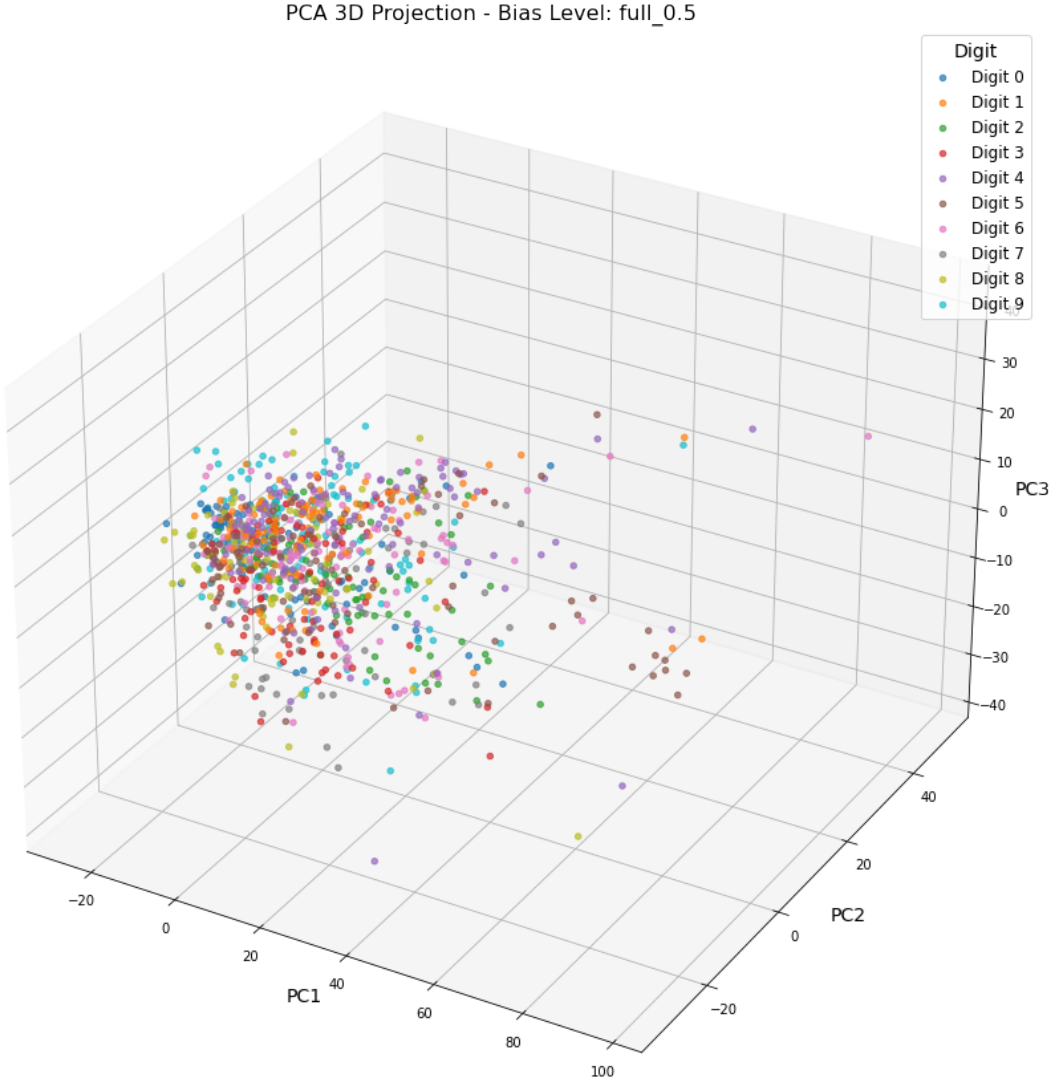
```
<Figure size 864x720 with 0 Axes>
```

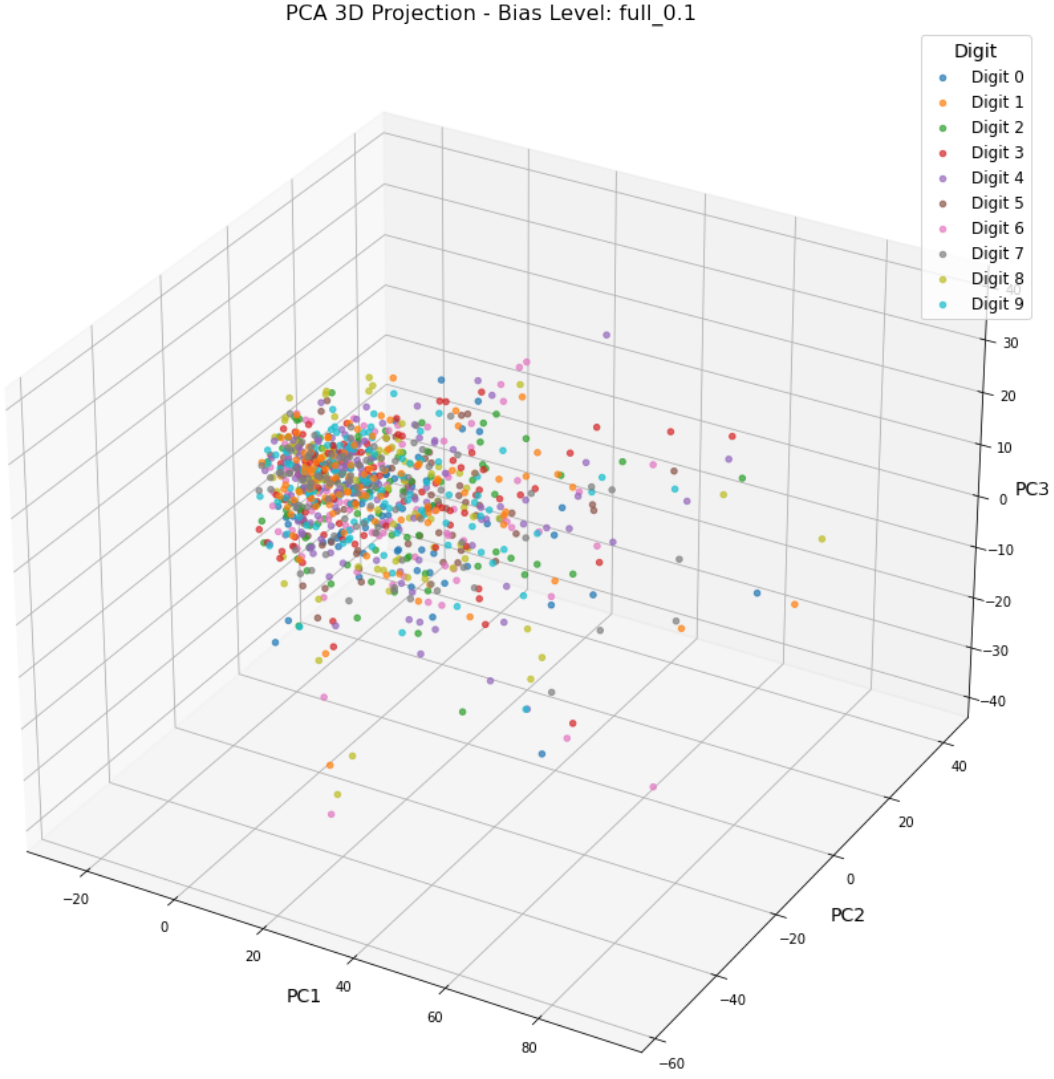


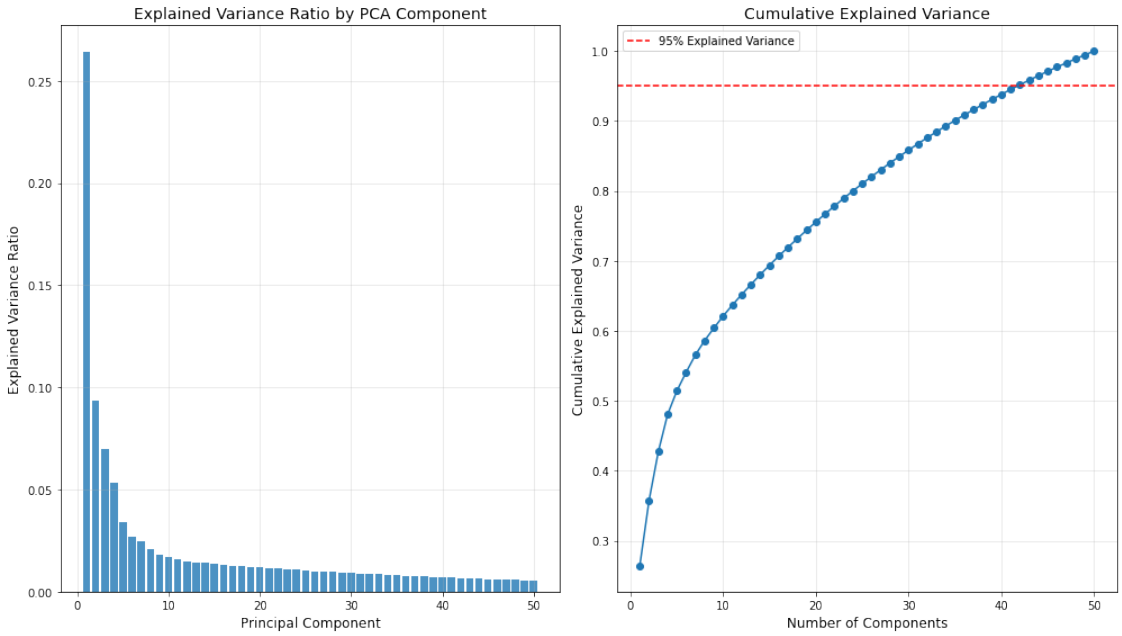

PCA 2D Projection - Bias Level: full_0.5
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
PC2
PC1
40
20
0
−20
−20
0
20
40
60
80
100


PCA 2D Projection - Bias Level: full_0.1
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
PC1
PC2
40
20
0
−20
−40
−60
−20
0
20
40
60
80


PCA 3D Projection - Bias Level: full_0.99
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
PC1
PC2
PC3


PCA 3D Projection - Bias Level: full_0.5
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
PC1
PC2
PC3

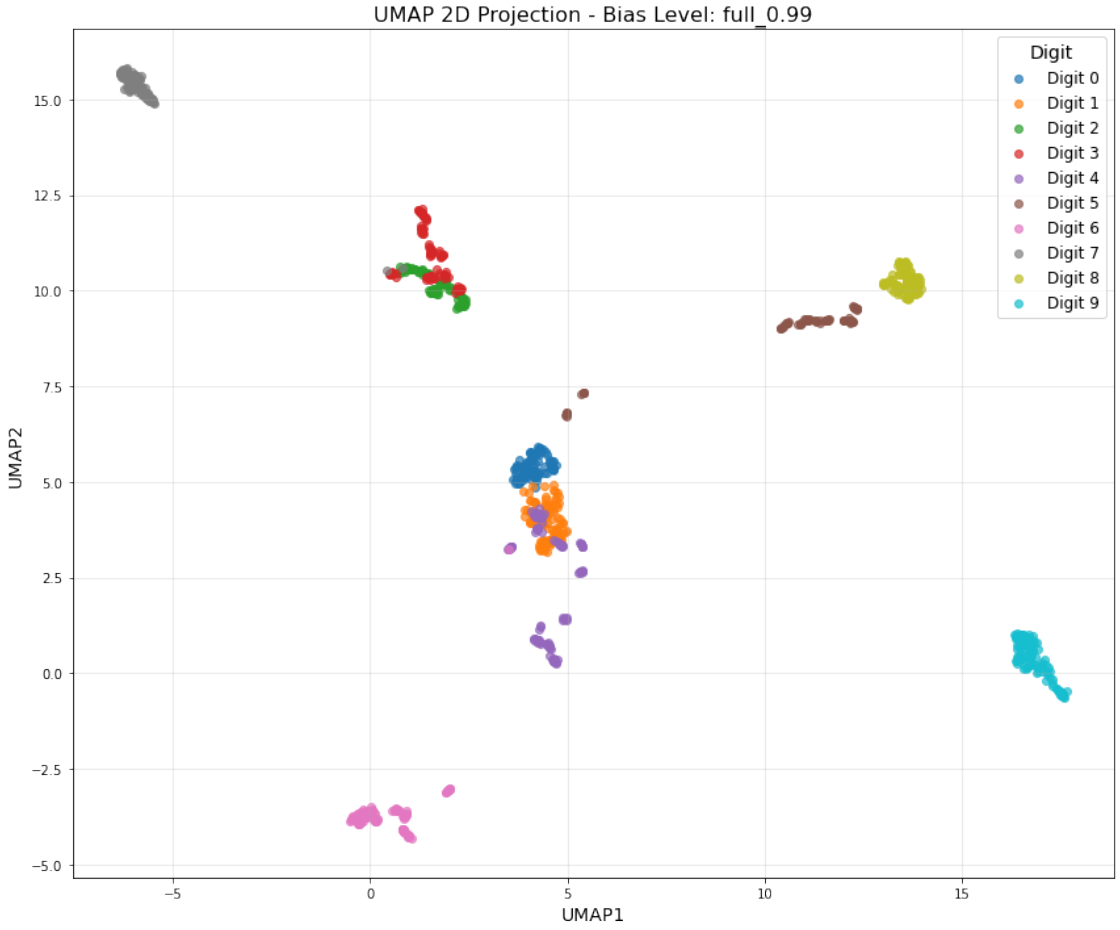

PCA 3D Projection - Bias Level: full_0.1
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
PC1
PC2
PC3



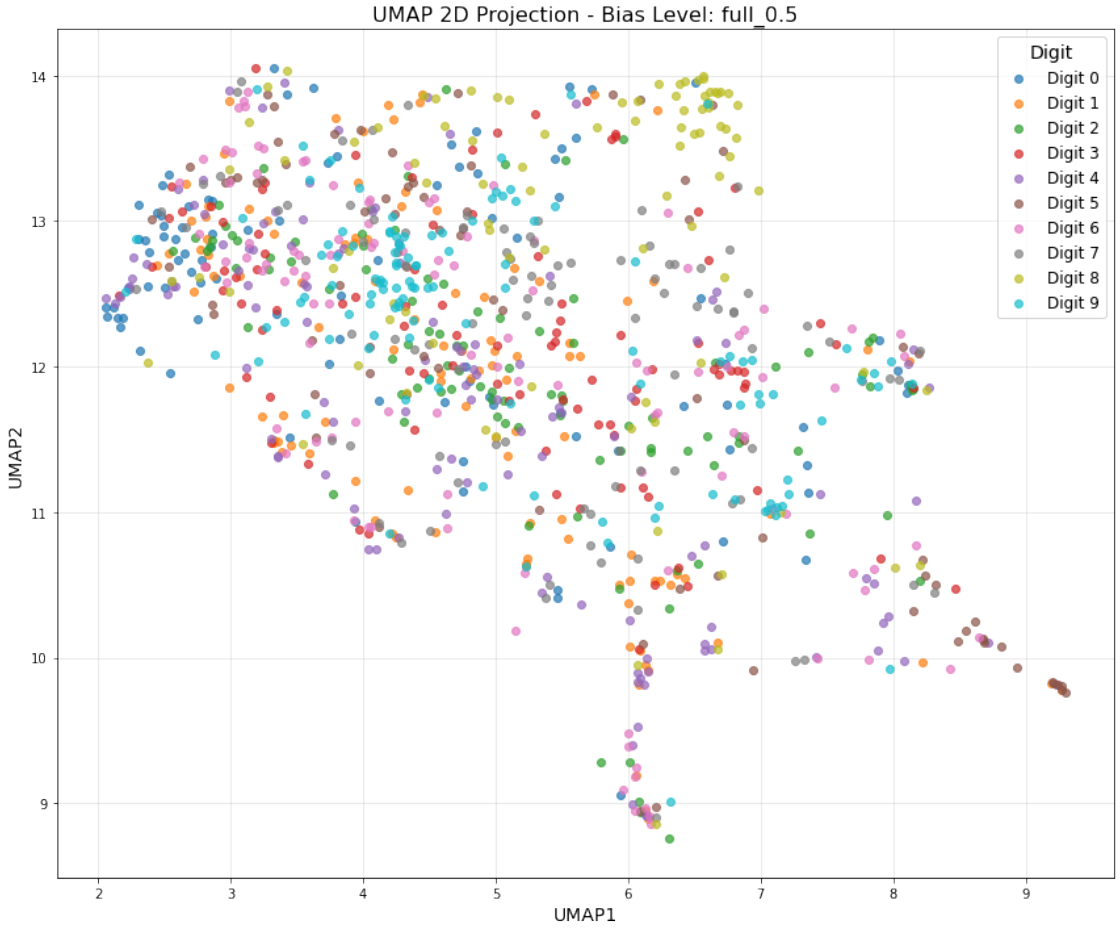
After performing PCA on each of the correlation datasets for MNIST, we can see that only the 0.99 dataset is clearly separable using PCA. Performing PCA on the 0.99 dataset yields distinct clusters of each digit, and this indicates that the majority of images for each digit look similar even when decomposed. This may be due to the fact that the majority of images with a certain digit have the same representation (same digit position, scale, color, texture etc) as we saw in the stacked bar charts above. The 0.5 and 0.1 datasets are not nearly as separable when using PCA.

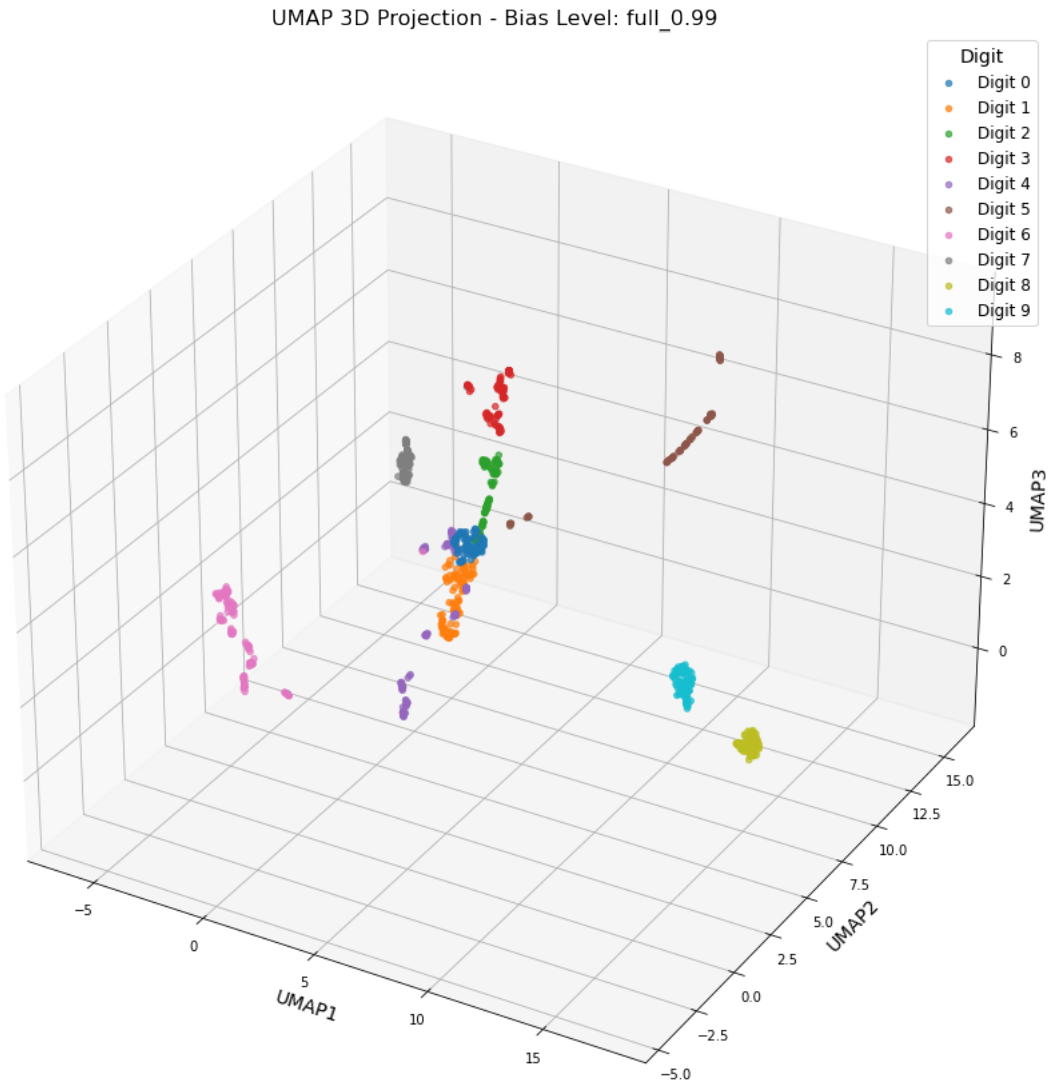
1.3.6 Dimensionality Reduction - UMAP

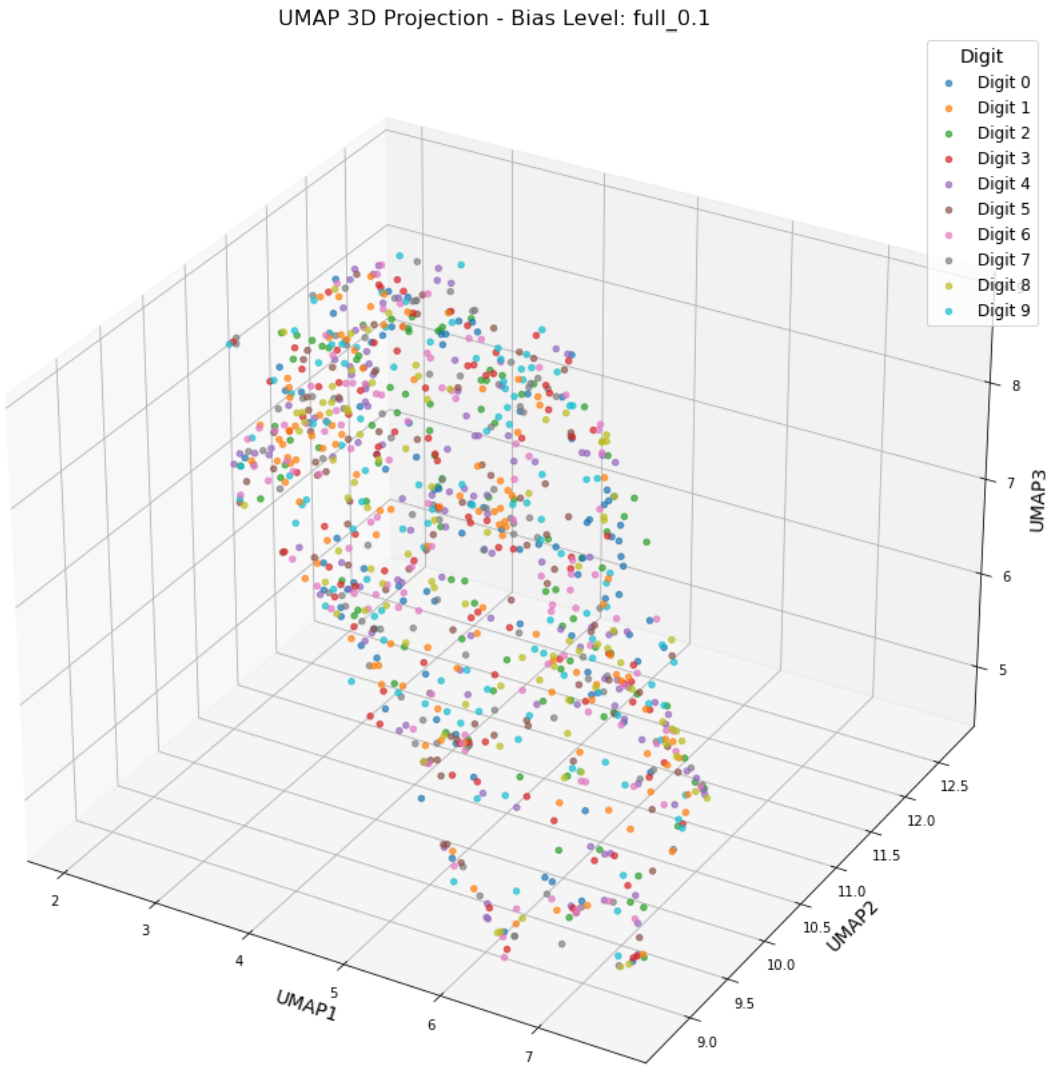
```
<Figure size 864x720 with 0 Axes>
```

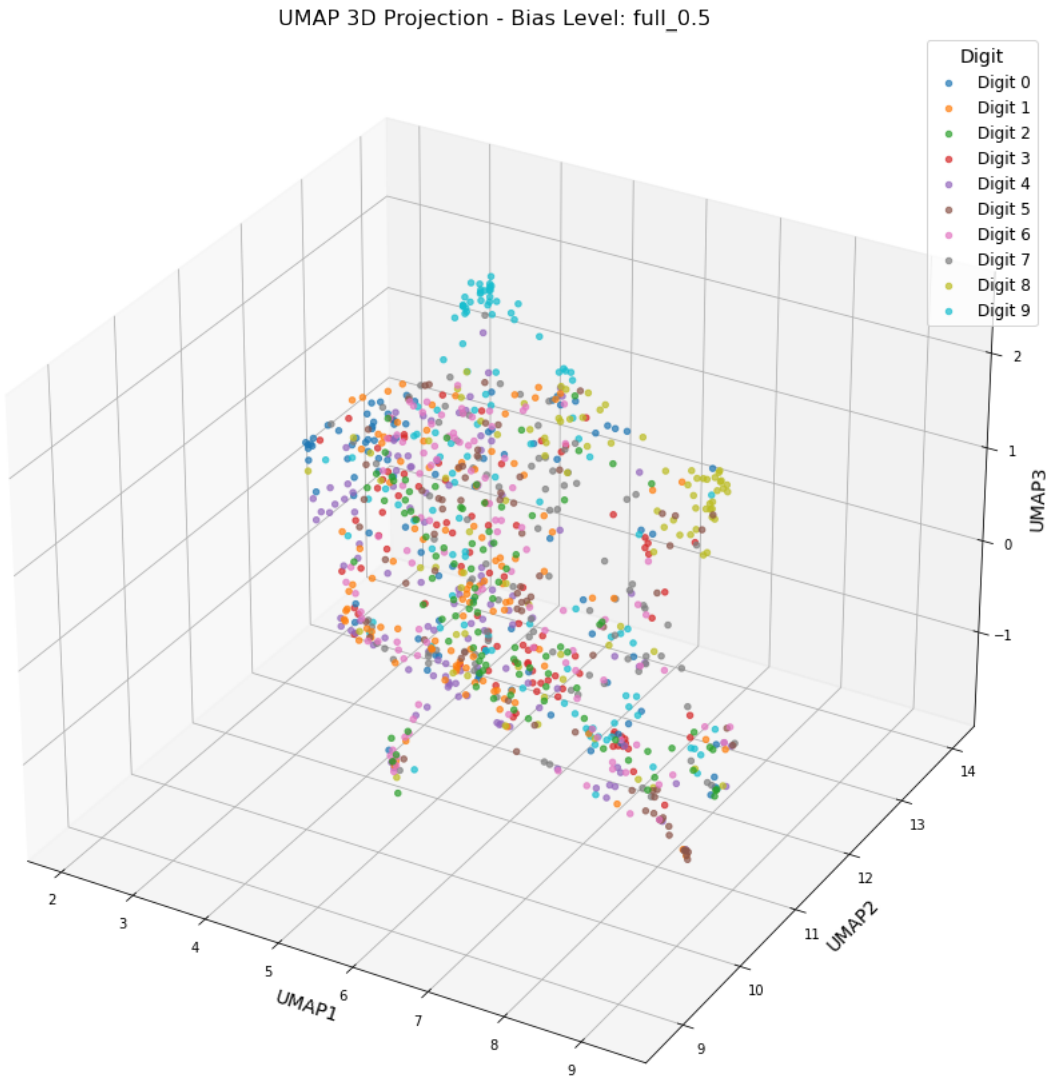

UMAP 2D Projection - Bias Level: full_0.99
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
UMAP2
UMAP1
15.0
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−5
0
5
10
15


UMAP 2D Projection - Bias Level: full_0.1
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
UMAP2
UMAP1
12.5
12.0
11.5
11.0
10.5
10.0
9.5
9.0
2
3
4
5
6
7


UMAP 2D Projection - Bias Level: full_0.5
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
UMAP2
UMAP1
14
13
12
11
10
9
2
3
4
5
6
7
8
9


UMAP 3D Projection - Bias Level: full_0.99
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
UMAP1
UMAP2
UMAP3

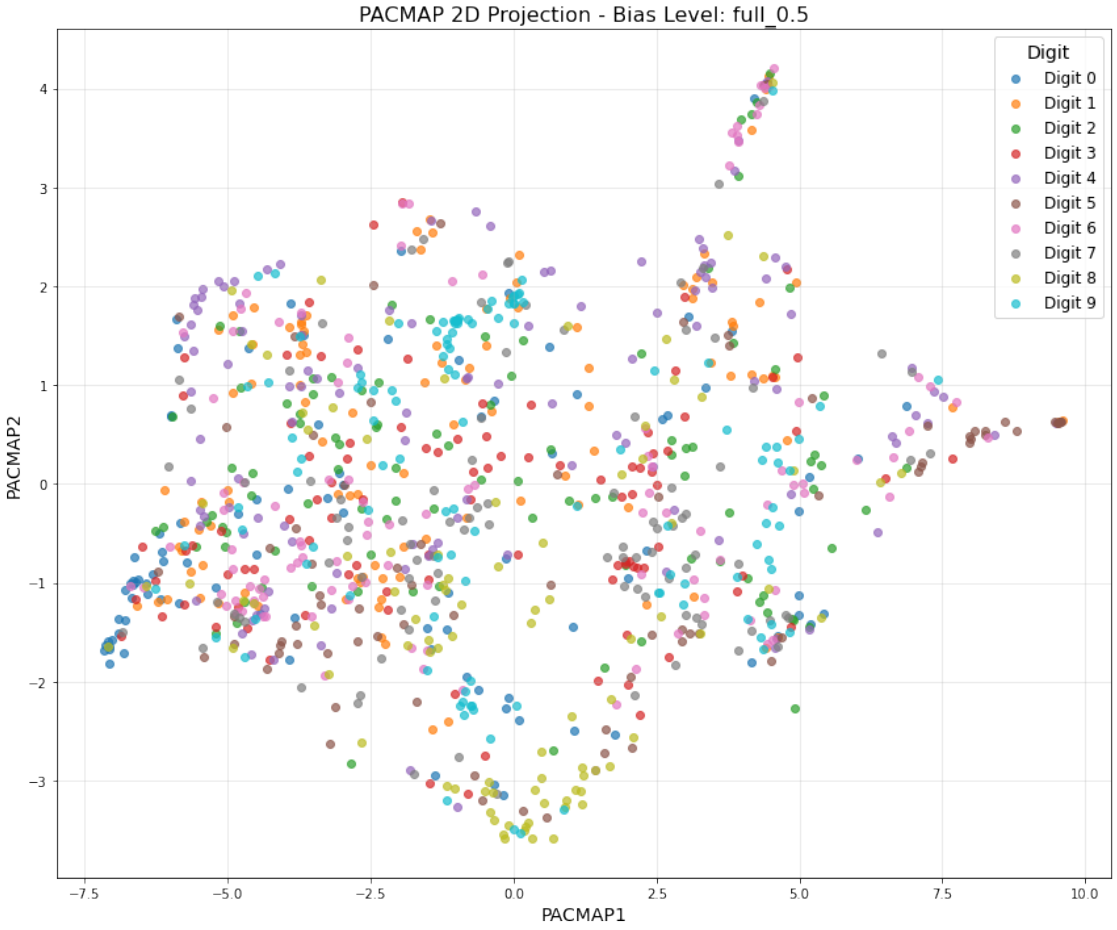

UMAP 3D Projection - Bias Level: full_0.1
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
UMAP1
UMAP2
UMAP3

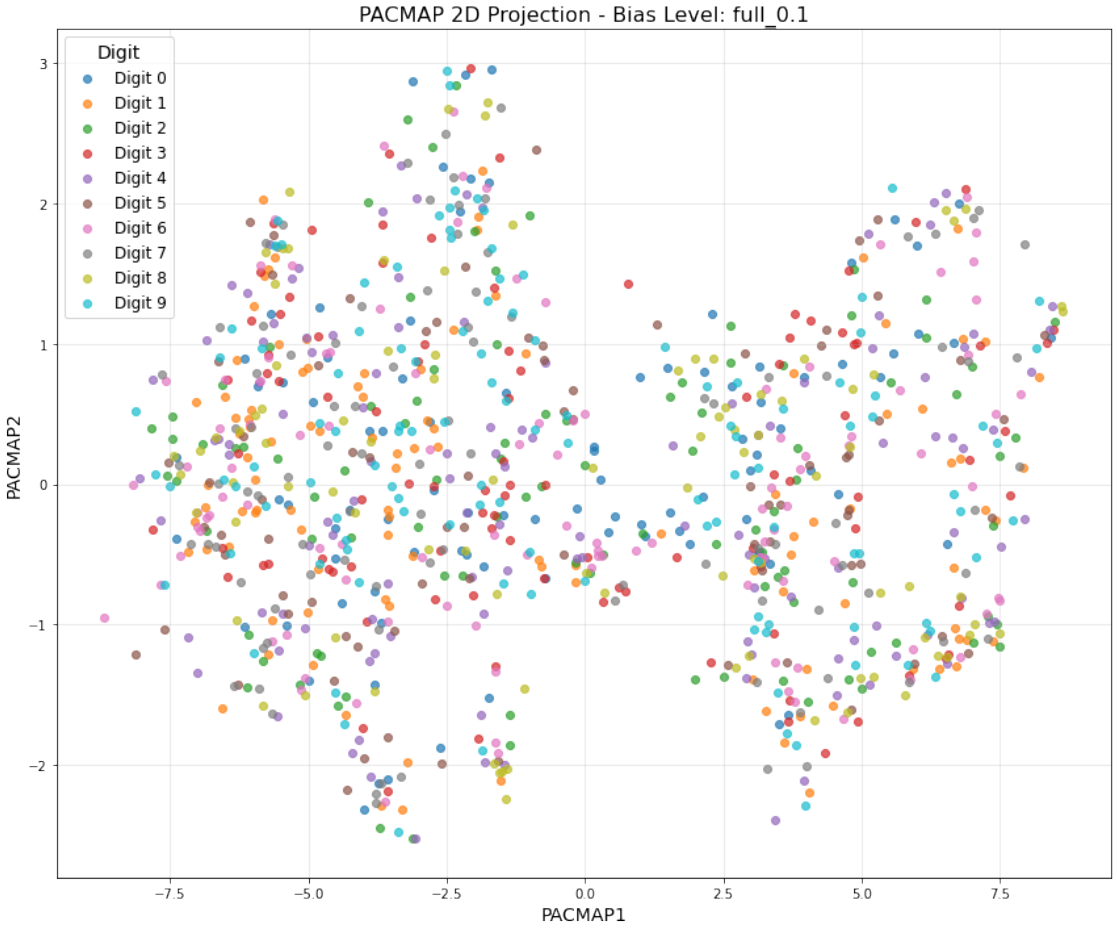


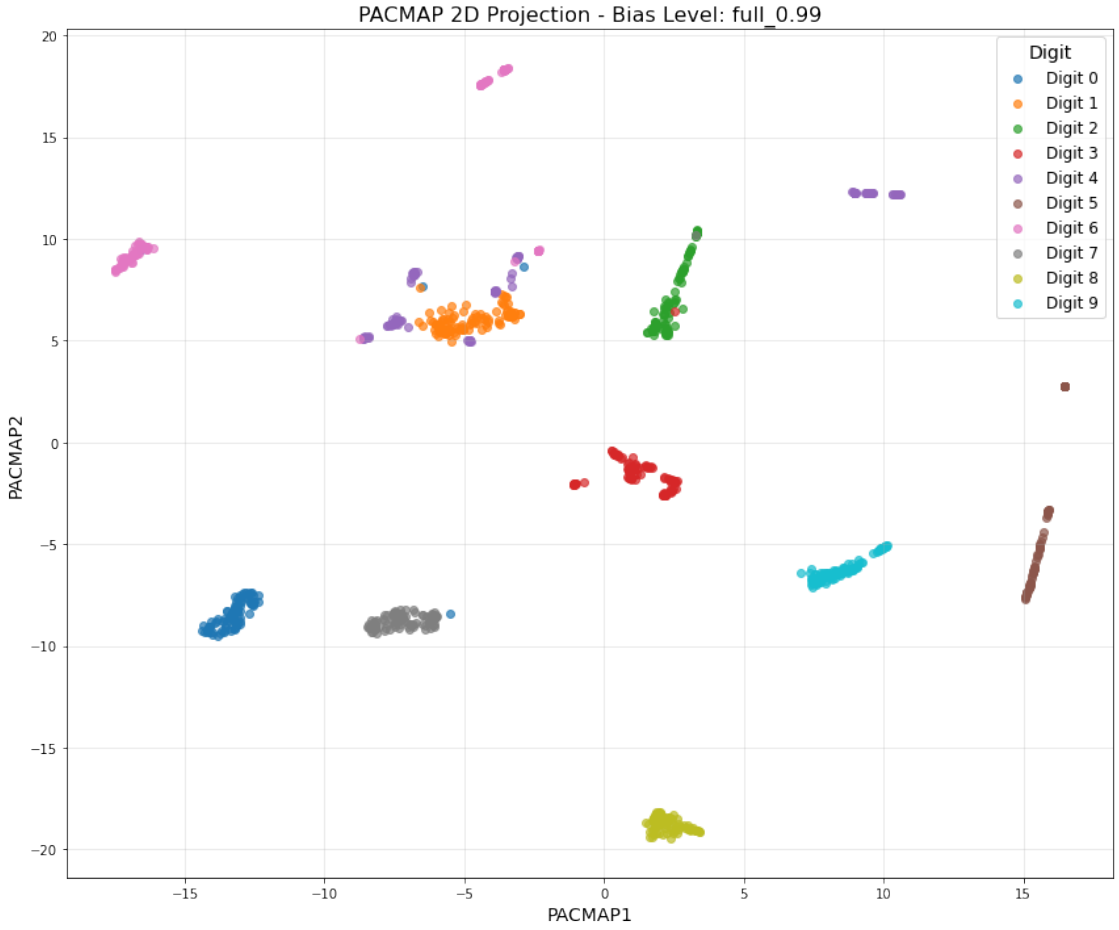
Dimensionality reduction with UMAP yields identical findings to those discovered when using PCA. Only the 0.99 dataset images are highly separable using this technique, while 0.5 and 0.1 dataset images are very difficult to separate using this dimensionality technique. Again, this is likely due to the high correlation of other variables with digit number in the 0.99 dataset, which results in the images of each digit looking very similar and making them easier to distinguish. From the visualizations, we can also see that digits 2 and 3 look very similar to each other, while 0, 1, and 4 also are difficult to distinguish when decomposed with UMAP.

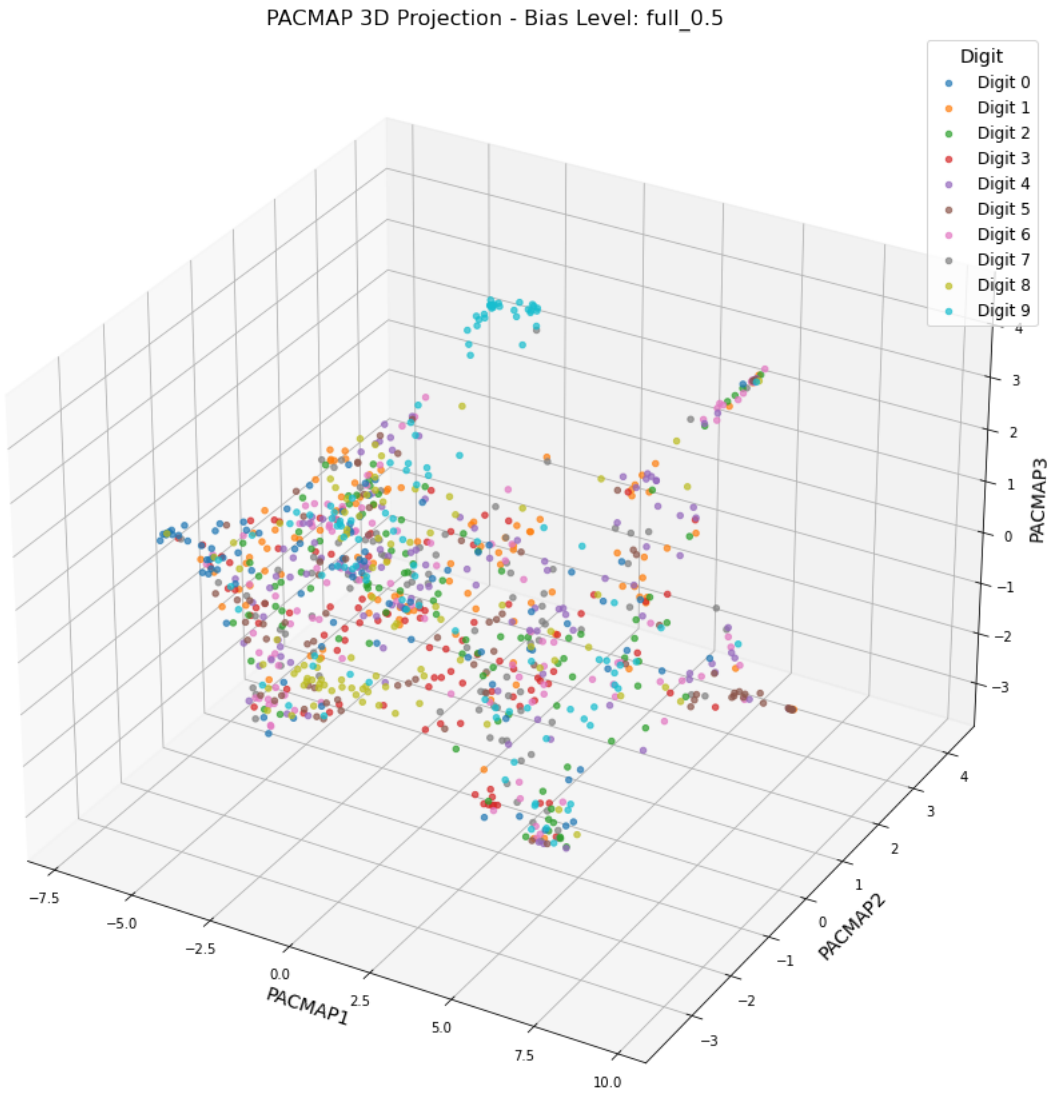
1.3.7 Dimensionality Reduction - PaCMAP

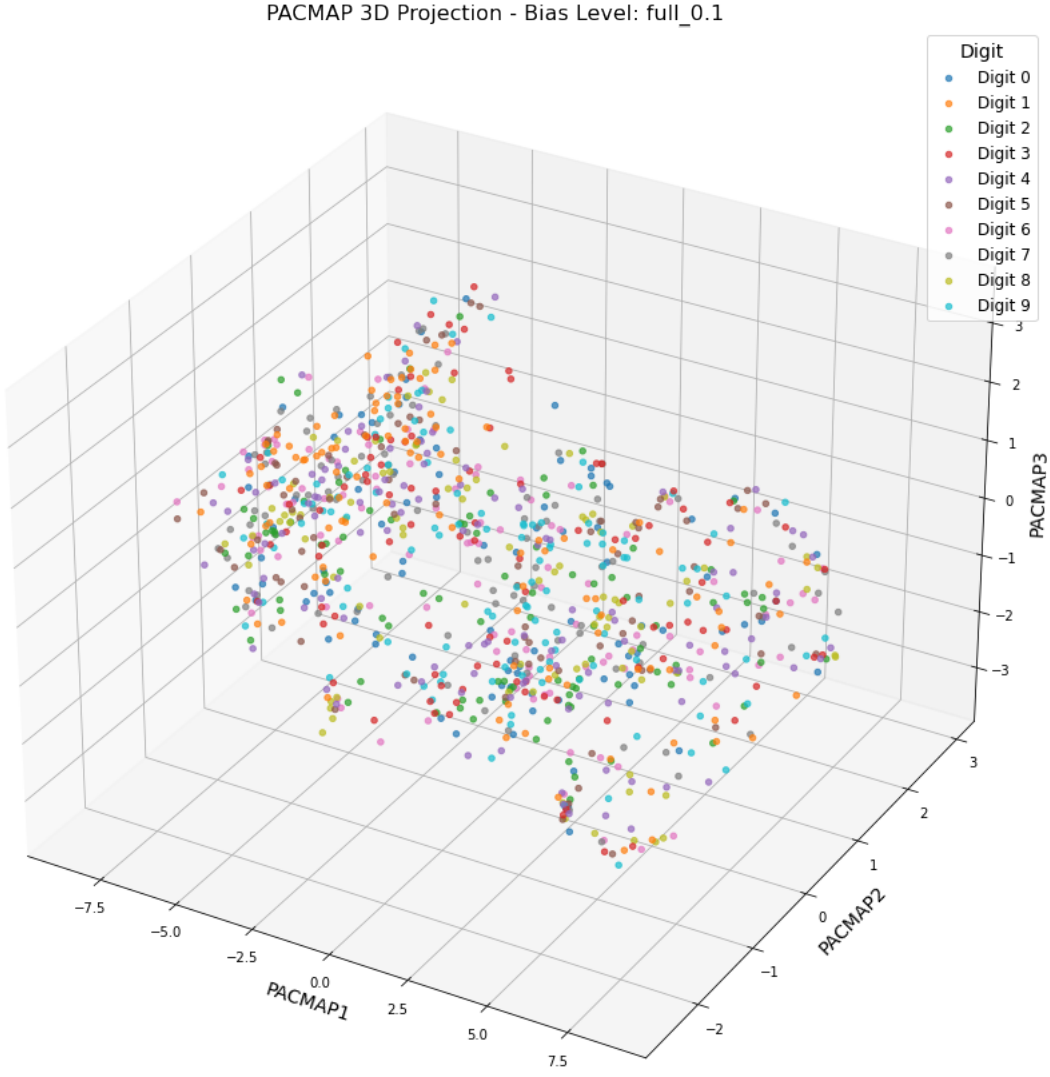
```
<Figure size 864x720 with 0 Axes>
```

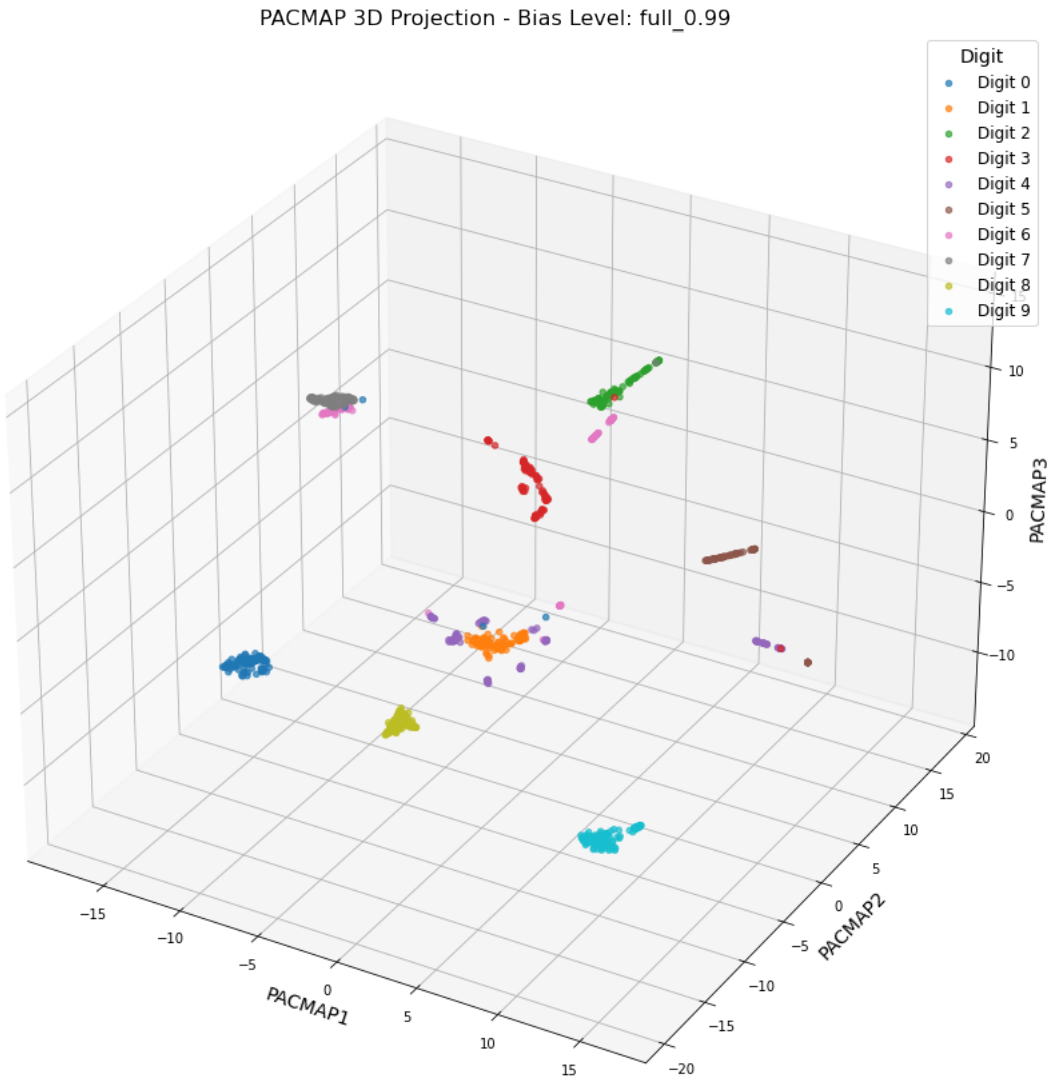

PACMAP 2D Projection - Bias Level: full_0.5
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
PACMAP2
PACMAP1
4
3
2
1
0
−1
−2
−3
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


PACMAP 2D Projection - Bias Level: full_0.1
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
PACMAP2
PACMAP1
3
2
1
0
−1
−2
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5


PACMAP 2D Projection - Bias Level: full_0.99
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
PACMAP1
PACMAP2
20
15
10
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
15


PACMAP 3D Projection - Bias Level: full_0.5
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
PACMAP1
PACMAP2
PACMAP3


PACMAP 3D Projection - Bias Level: full_0.1
Digit
Digit 0
Digit 1
Digit 2
Digit 3
Digit 4
Digit 5
Digit 6
Digit 7
Digit 8
Digit 9
PACMAP1
PACMAP2
PACMAP3



Again, our findings using the previous two dimensionality reduction techniques are supported using PaCMAP. The 0.99 dataset is again very separated, while the 0.5 and 0.1 datasets are not. The 0.5 dataset shows some additional separation between digits, although it is not very much. The separation between digits also appears to be the highest out of all three dimensionality reduction techniques using PaCMAP. PaCMAP indicates that 6 and 7 look similar when decomposed, as well as 4 and 1.

```
[NbConvertApp] Converting notebook T2P4_Exploratory_Data_Analysis.ipynb to pdf
[NbConvertApp] Support files will be in T2P4_Exploratory_Data_Analysis_files/
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
```

```
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Making directory ./T2P4_Exploratory_Data_Analysis_files
[NbConvertApp] Writing 47335 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 4472916 bytes to
/Users/leanne/Documents/WSU/DATA424/T2P4_Exploratory_Data_Analysis.pdf
```