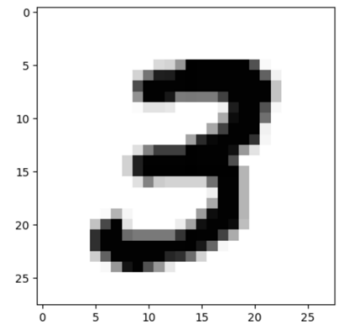
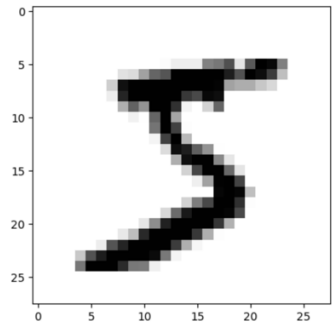
Group LASSO Multinomial Regression Report Machine Learning
By Dimitris Markopoulos Columbia University

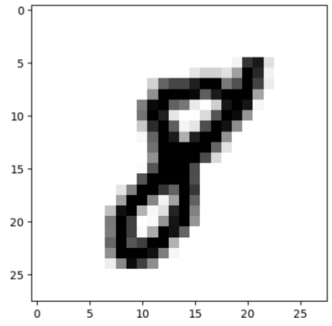
Group Lasso is particularly effective for this MNIST subset (digits 3, 5, 8) because it enables structured feature selection by penalizing groups of features together. Unlike standard Lasso which treats pixels individually across each class, Group Lasso can leverage the spatial coherence in handwritten digits by treating neighboring pixels as related groups, preserving important structural information. This approach is especially valuable for distinguishing between the visually similar digits 3, 5, and 8, as it can identify which regions of the image (such as the upper loops or middle strokes) are most discriminative across all classes. By automatically selecting relevant pixel groups while eliminating irrelevant ones, Group Lasso would reduce model complexity, improve interpretability by highlighting the most important digit regions for classification, and potentially enhance generalization by preventing overfitting to noise in the training data. The technique strikes an optimal balance between the flexibility of considering all pixels and the efficiency of dimensionality reduction, making it well-suited for image classification tasks where features have natural groupings.



*** Image is classified as a 3 ***



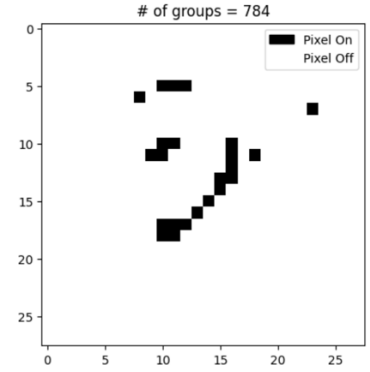
*** Image is classified as a 5 ***



*** Image is classified as a 8 ***

As we can see there exists pixels that are far more relevant in classifying a digit then others. For example, the pixels in the corners of the image have no significance in classifying the image across all classes; group LASSO multinomial regression will seek to shrink features that are not significant across all classes to zero and perform feature selection. Now the question is, how do we perform grouping? I have performed three different grouping methods to compare with each other.

The first grouping method was to group a single $1x1$ pixel across all methods. The results yielded significant features (pixels) across all methods. I have plotted these features below; assume the image is 28×28 given by $\begin{bmatrix} 1 & 2 & \cdots & 28 \\ 29 & 30 & \cdots & 56 \\ \vdots & \vdots & \ddots & \vdots \\ 757 & 758 & \cdots & 784 \end{bmatrix}$.




```
Accuracy: 0.8084
Classification Report:
              precision    recall  f1-score   support

           3       0.77      0.87      0.82      1426
           5       0.80      0.76      0.78      1257
           8       0.86      0.79      0.82      1373

    accuracy                           0.81      4056
   macro avg       0.81      0.81      0.81      4056
weighted avg       0.81      0.81      0.81      4056
```

I then fit and visualized 2×1 and 2×2 grouping below:

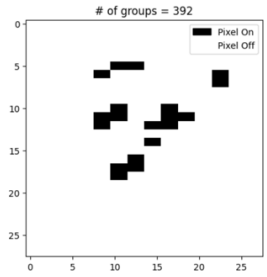
```
Accuracy: 0.8151
Classification Report:
              precision    recall  f1-score   support

           3       0.78      0.88      0.83      1426
           5       0.81      0.76      0.78      1257
           8       0.86      0.81      0.83      1373

    accuracy                           0.82      4056
   macro avg       0.82      0.81      0.81      4056
weighted avg       0.82      0.82      0.81      4056

There are 38 nonzero features after applying group LASSO.
These are the pixels group LASSO found to be significant in classification of the digit to either 3, 5, or 8.

Pixel List = ['pixel151', 'pixel152', 'pixel153', 'pixel154', 'pixel177', 'pixel178', 'pixel191', 'pixel192', 'pix
el219', 'pixel220', 'pixel291', 'pixel292', 'pixel297', 'pixel298', 'pixel317', 'pixel318', 'pixel319', 'pixel32
0', 'pixel325', 'pixel326', 'pixel327', 'pixel328', 'pixel345', 'pixel346', 'pixel351', 'pixel352', 'pixel353', 'p
ixel354', 'pixel407', 'pixel408', 'pixel461', 'pixel462', 'pixel487', 'pixel488', 'pixel489', 'pixel490', 'pixel51
5', 'pixel516']
```



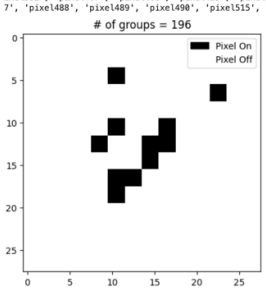

```
Accuracy: 0.8074
Classification Report:
              precision    recall  f1-score   support

           3       0.77      0.87      0.82      1426
           5       0.82      0.74      0.78      1257
           8       0.84      0.81      0.82      1373

    accuracy                           0.81      4056
   macro avg       0.81      0.80      0.81      4056
weighted avg       0.81      0.81      0.81      4056

There are 44 nonzero features after applying group LASSO.
These are the pixels group LASSO found to be significant in classification of the digit to either 3, 5, or 8.

Pixel List = ['pixel123', 'pixel124', 'pixel151', 'pixel152', 'pixel191', 'pixel192', 'pixel219', 'pixel220', 'pix
el291', 'pixel292', 'pixel297', 'pixel298', 'pixel319', 'pixel320', 'pixel325', 'pixel326', 'pixel345', 'pixel34
6', 'pixel351', 'pixel352', 'pixel353', 'pixel354', 'pixel373', 'pixel374', 'pixel379', 'pixel380', 'pixel381', 'p
ixel382', 'pixel407', 'pixel408', 'pixel435', 'pixel436', 'pixel459', 'pixel460', 'pixel461', 'pixel462', 'pixel48
7', 'pixel488', 'pixel489', 'pixel490', 'pixel515', 'pixel516', 'pixel543', 'pixel544']
```



The accuracy is good across all grouping methods with larger groups having significantly more efficient runtimes. Further improvements can be done through hyperparameter tuning to optimize the parameters group_regularization and l1_regularization (as well as group selection like what we have done here but on a more technical level instead of intuitively fitting groups). I would argue that at this point runtime does become an issue on cpu and efficiency matters to me (as the first method took 8 minutes to run). Now if we had to further apply hyperparameter tuning it becomes unreasonable to fit with such large groups so a 2×2 grouping for insignificant reduction in accuracy is appealing.