Why was 6 afraid of 7? Because 7 8 9!: Feature Engineering MINST Handwriting Data to Classify 6s and 7s

By Piper Dean and Marc Eidelhoch

Introduction:

In machine learning, sometimes you need to manipulate your predictor variables in order to create a model to predict a given outcome. The building of these predictors is referred to as feature engineering. Once created, these predictors are used within a model to classify data and predict an outcome. The goal of our project was to build a model that allowed us to predict whether a given handwritten digit was a six or a seven. To do this, we used feature engineering to build two different regions of interest and used the proportion of dark pixels in each region to classify a digit as likely a six or a seven.

Data:

The data for the handwritten analysis is from the Modified National Institute of Standards and Technology (MNIST) database. The database contains thousands of digitized images of handwritten digits from zero to nine. Since our analysis focuses on sixes and sevens, we pulled the 12,183 observations of the two digits which contain 5,918 sixes and 6,265 sevens. Each observation or row in the data frame contains 785 variables. The first column, "digit", is our response variable that denotes the actual number that is written. The other 784 variables are predictors which correspond to a location in a 28 by 28 matrix. Each predictor variable contains a value from 0 to 255 which corresponds to the gradient of that location. A zero is a white background and 255 is a black pixel. Thus, if you plot each of the grayscale values of the 784 predictor variables you can see the handwritten number. We were able to do this in R using a function given by Adam. You can see in Figure 1 an example of a handwritten six and a handwritten seven from the dataset.

Regions of Interest:

To build our model, we decided to take the 784 predictor variables in the dataset and generate predictor variables for our model by selecting regions of interest and calculating the proportion of dark pixels in each region. We decided that one region would be the points that likely belong to a 6 and the other would be points that likely belong to a 7. After splitting the data into training and testing sets, we used our training data to create an average version of each digit. We did this by adding together the value at each pixel across all of our sixes and similarly across all of our sevens. This gave us one 28 by 28 matrix for each number where we could determine what points were most common for each digit across all of our observations and plotted this heatmap (Figure 2). Each image emphasizes the areas of importance where the darker location has more overlap between individual observations. Then, for each pixel, we took the value of the sixes minus the sevens which gave us the difference between the two digits. For a given pixel, a value close to zero means that there are about as many dark pixels in a six or a seven so that pixel is probably not a great predictor of the digit. However, a large number means that this pixel is dark for a lot of sixes but not for a lot of sevens and a very negative number means the opposite. Thus, in Figure 3 the darker areas show the unique locations of sixes while the lighter regions represent the unique locations of sevens.

To create our actual predictors, we defined the "unique" locations by a threshold of 400,000 as the sum of all values at that pixel location from our aggregate data frame. There are 57 unique locations for six and 37 unique pixel locations for seven shown in Figure 4. These groupings of unique pixel locations become our two regions of interest. We then calculate the proportion of dark pixels within the region of interest for each observation which are our two predictor variables. The boxplot (Figure 5) shows the proportion of dark pixels from a region grouped by digit. The graph reflects what we expected to see, with a higher proportion of dark pixels from the digit seven in the seven's region of interest and a lower proportion of pixels of sevens in the sixes region of interest and the same is true for sixes.

Model:

After building our predictors for these two regions of interest, we focused on three different classification models: quadratic discriminant analysis (QDA), linear discriminant analysis (LDA), and logistic regression. We did not consider LASSO or RIDGE regression because we only had two predictors so we didn't need to make big variable selection decisions. Our goal in our analysis was to build a classifier that accurately predicts whether a digit is a six or a seven, so we chose accuracy as the metric to assess our model's performance. As mentioned earlier the data was split into training and test data with 75% of the data used to train the model. We then used k-fold cross-validation with ten folds on our training data to compare the three models. Ten-fold cross-validation relies on splitting the training data into ten folds, training the model on nine of the folds, and holding out one fold for testing. This process is then repeated ten times, each time holding out a different fold. After each iteration, we calculated the accuracy of the test data and then averaged the accuracy across all ten iterations for each model type[1]. We selected the model with the highest cross-validated classification accuracy to fit as the final model.

Results:

After cross-validation, the three models yielded an aggregate accuracy of 98.91% for QDA, 98.86% for LDA, and 98.89% for logistic regression. Thus, we chose QDA as our final model. With QDA, our model is less interpretable than it would be with something like logistic regression. However, we are primarily concerned with accurate predictions so having a less interpretable model doesn't matter. Second, after plotting the data it appears to be a better fit to non-linear bounds (Figure 6), so we would think QDA would be more accurate than using linear

[1] Resource for building cross validation for-loop in R
https://stackoverflow.com/questions/75548116/resamples-folds-for-cross-validation-in-r

decision boundaries. The final QDA model was 98.88% accurate when classifying the test data. Additionally, the sensitivity (true positive rate) and specificity (true negative rate) are 98.99% and 98.79% respectively.

Since the goal of this analysis was to classify sixes and sevens, our model is highly specified to those two digits. Therefore, a limitation of our analysis is that we cannot generalize this model to other digits. To generalize our model to all digits or characters, we would have to reform our classifiers to be more generalized. For example, we could use a quadrant approach, like the x-y coordinate plane, and create regions of interest as the four quadrants. Another limitation of our analysis is that it is very dependent on all of the digits being approximately the same size and in approximately the same place in the matrix. If this is not the case, we would also need to consider a different approach to classifying the digits or more preprocessing.

Appendix:

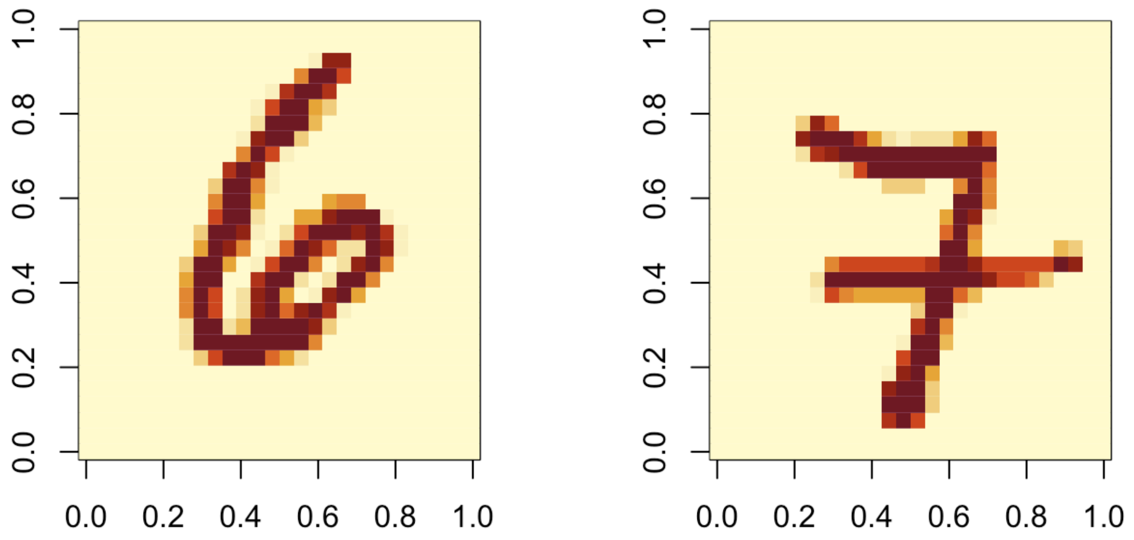

Figure 2: Visual of a six and a seven

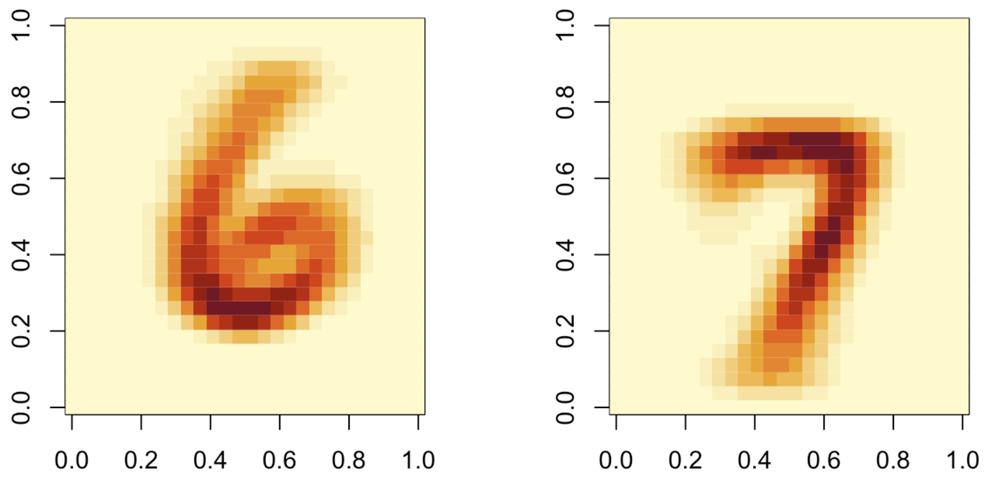

Figure 2: Aggregate plotting of all sixes and sevens

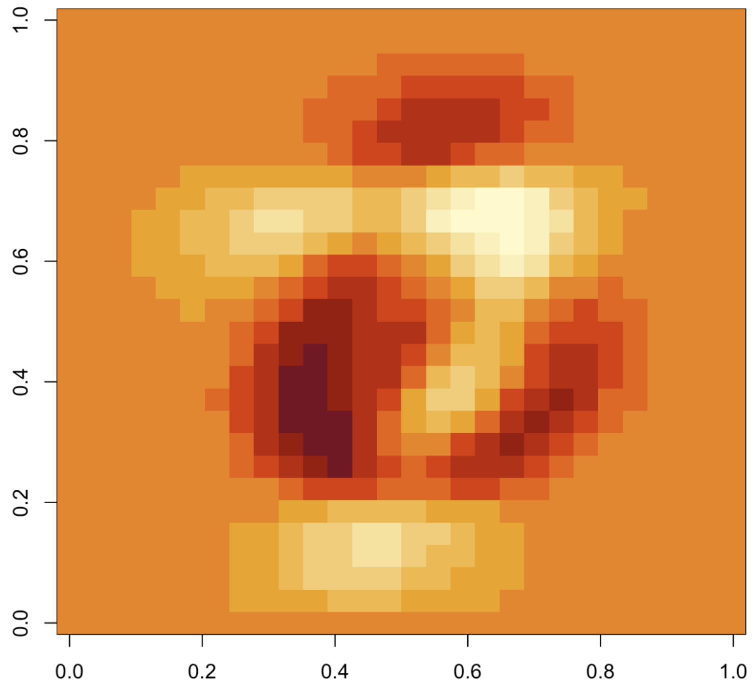

Figure 3: Aggregate sixes and sevens juxtaposed against each other

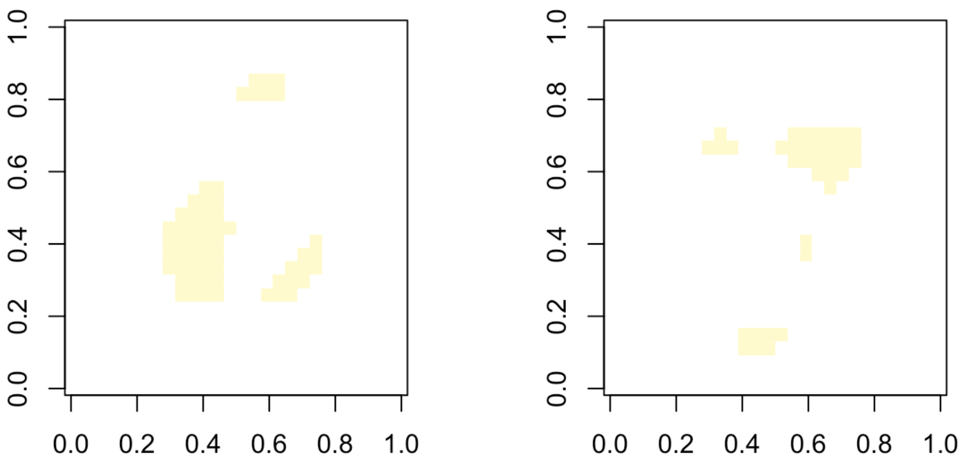

Figure 4: Regions of interest for six and seven

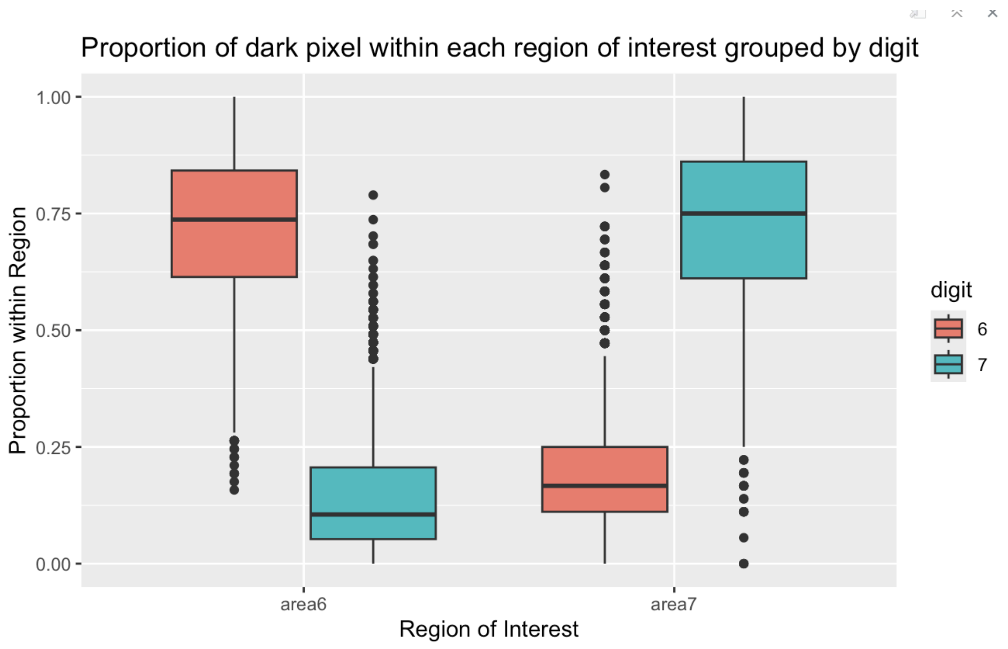

Figure 5: Boxplot comparing the proportion of pixels in the region of interest grouped by digit for training data

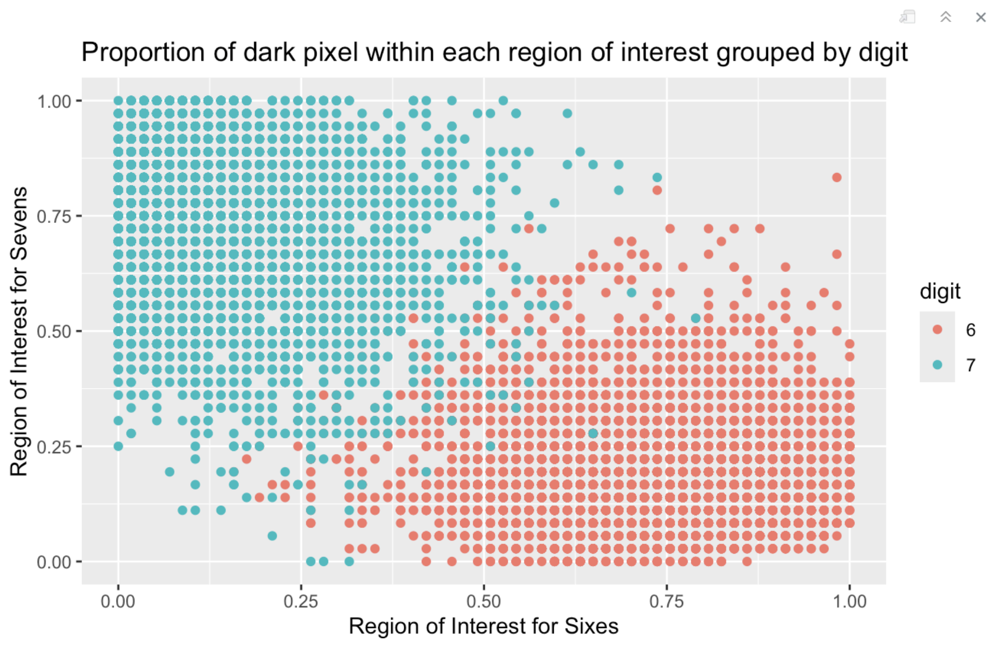

Figure 6: Scatter plot of the proportion of pixels in the region of interest group by digit for training data