I began this process by remembering that in Assignment 3 we had investigated PCA as a possible route for reducing dimensions to improve performance of the classifiers we used. Noticing that in the case of trying to classify all digits PCA had limited benefits – I decided to preserve time and not utilize any PCA in order to spend more time performing grid searches using high-performing classifiers. The classifiers I attempted to leverage in this assignment were 3 ensemble methods (bagging, boosting, and random forest) and a support vector classifier. From the intuition I’ve gathered from class and experience it seems that without venturing into the realm of neural networks these 4 methods tend to have the highest-performing results. Now that I had these 4 algorithms defined I need to tweak them to find the hyper-parameters that yield the best results. I started this process by performing a general gridsearch on each method to try to find the best hyperparameters to maximize accuracy. I used a 3-fold cross-validation for this process. While 3-fold cross-validation is not ideal for yielding reliable scoring for each candidate (3 is a small sample size) I had decided to start with this small number because initially I had tried a 30-fold cross validation and noticed the training time for these large grids would last weeks to yield results for each algorithm. As a compromise I decided to start with 3-fold for the general hyper-parameter gridsearch to yield the best hyperparameters for each algorithm – then enter another round of gridsearches (with smaller grids) and use 30-fold cross-validation to report the final results. The idea behind the 2x gridsearch method is the first gridsearch quickly gives you an idea of where you should be investigating ideal hyperparameters – then when you have “trimmed the fat” on the grid use 30-fold cross-validation to compare each algorithm to each other.

To perform this gridsearch I needed to binarize the labels of the output and apply a one-vs-rest classifier approach to handle all outputs of the classifier. One-vs-rest classifier means one classifier is fit per-class and for each classifier the class is fitted against all the other classes. With each of the 4 selected algorithms wrapped in this one-vs-rest classifier I was able to perform a gridsearch that optimized accuracy on the results of each class in each algorithm.