Binary Classification Using Logistic Regression and Random Forests

Cameron Wat 12/15/2023

Abstract

In this part of the project, we will explore binary classification using logistic regression and random forests on the MNISTmini dataset. Logistic regression is primarily used for binary classification problems, where the outcome is either 0/1 or yes/no. Random Forests are an ensemble learning method used for classification and regression tasks. The MNISTmini dataset is a subset of the MNIST containing grayscale images of handwritten digits (0-9). The objective is to perform binary classification, to distinguish two assigned digits, in our case, the numbers 5 and 9.

Methodology

We first loaded our dataset using the SciPy python library. Next, we normalized the pixel values so that they fall between the ranges of [0, 1]. We did this by dividing the pixels by 255.

Then we defined our classes we wanted to perform binary classification. Our classes were 5 and 9.

We limited our sample size to 1500, and split our dataset into 3 groups, the training set, validation set, and the test set. Each of these sets contained 500 data points.

To begin the regression model training, we used a range of strengths for our C_values (10^-7 to 10^5). Our test size was 0.2, and this represents the proportion of the dataset to include. Our random state was 42 and that initializes the random number generator, which decides the splitting of data into train and test sets.

For our random forest model, we trained it on the following n_estimators: [50, 100, 150, 200, 1000]. These are the number of trees in the forest. Like the regression method, we also used 42 as our random state.

Results

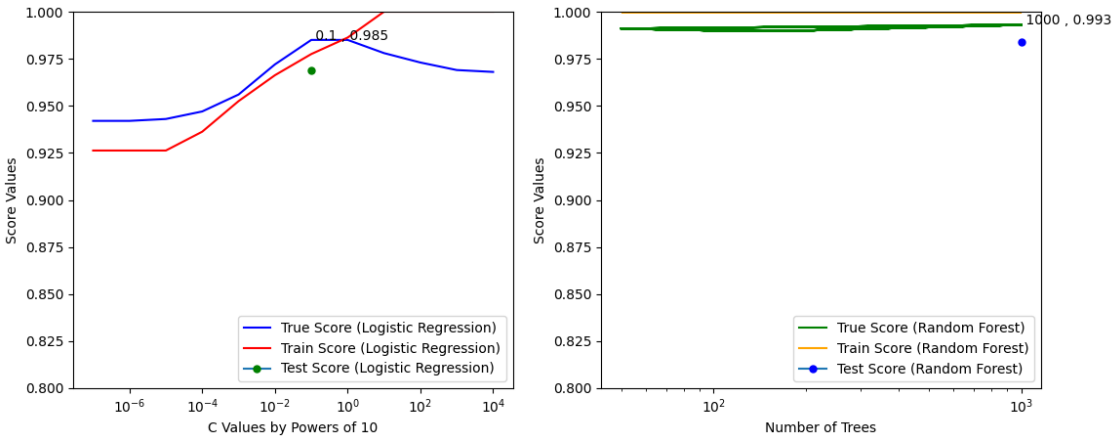

Figure 1: Logistic Regression (left) and Random Forest (right) results

From Figure 1, the Train Score for the Logistic Regression model started increasing once our C_values reached 10^-4. Our True Score peaked at 0.985 (98.5%) and our Test Score was 0.969 (96.9%). Overall very good scores.

For our Random Forest model, the Train Score was 1.000 (100%) through all the trees. The Train Score, while 1.000, is unlikely. This would suggest our model is flawless and was able to classify the numbers 100% of the time. Which is unlikely and might be an indication of an error in the code. Our True score was very close, and reached 0.993 (99.3%). The Test Score reached 0.984 (98.4%). Still very good scores. After running the code a few more times, the Train Score was 1.000 (100%) through all the trees. Which is surprising.

Logistic Regression Score against the test data: 0.969
Random Forest Score against the test data: 0.984

Conclusion

Based on the results above our Random Forest did better than our Logistic Regression by 0.015 (1.5%). Both achieved high accuracy when classifying the numbers 5 and 9 from the MNISTmini dataset. Our methods when creating both the Random Forest and Logistic Regression produced high results for both.