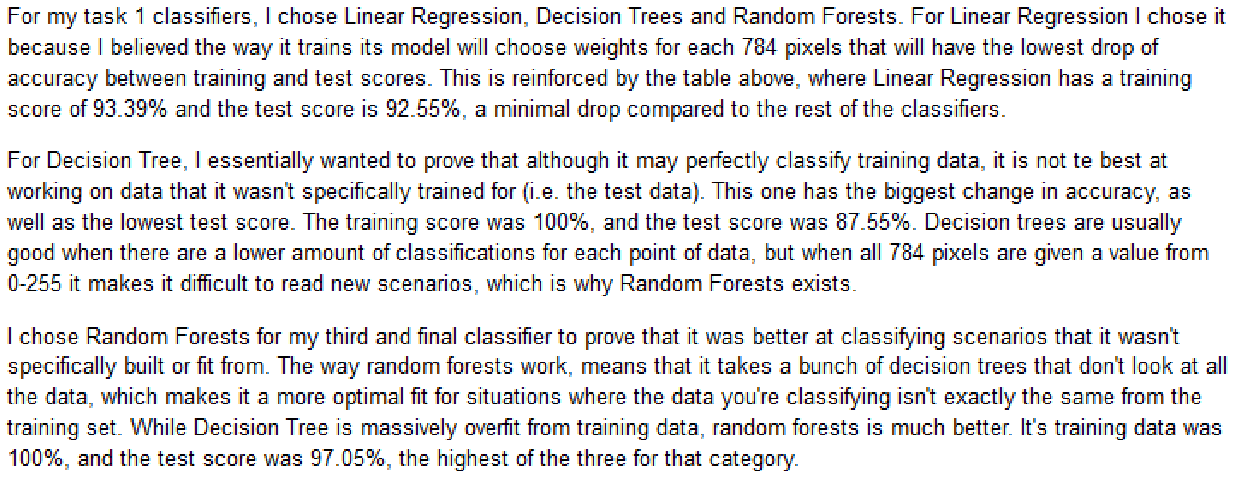
Jon Cucci

CS559 – Fundamentals of Machine Learning

12/16/21

Final Report

For this final project, we were asked to use the MNIST784 dataset to train assorted Machine Learning Classifiers, score them, and compare them. For task #1 we had to choose three non-deep learning models, and I chose Logistic Regression, Decision Tree, and Random Forest. I wrote a description in my Jupiter notebook about why I chose them, and what their strengths and weaknesses are.



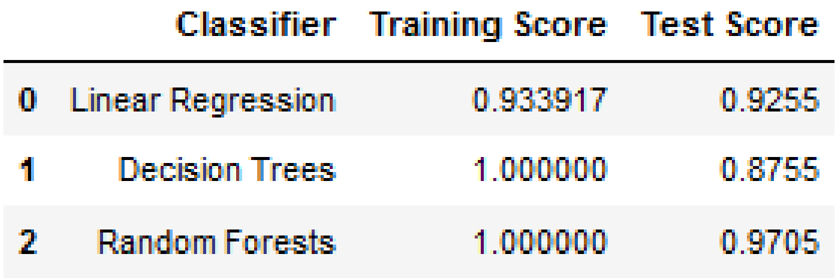
For my task 1 classifiers, I chose Linear Regression, Decision Trees and Random Forests. For Linear Regression I chose it because I believed the way it trains its model will choose weights for each 784 pixels that will have the lowest drop of accuracy between training and test scores. This is reinforced by the table above, where Linear Regression has a training score of 93.39% and the test score is 92.55%, a minimal drop compared to the rest of the classifiers.

For Decision Tree, I essentially wanted to prove that although it may perfectly classify training data, it is not te best at working on data that it wasn't specifically trained for (i.e. the test data). This one has the biggest change in accuracy, as well as the lowest test score. The training score was 100%, and the test score was 87.55%. Decision trees are usually good when there are a lower amount of classifications for each point of data, but when all 784 pixels are given a value from 0-255 it makes it difficult to read new scenarios, which is why Random Forests exists.

I chose Random Forests for my third and final classifier to prove that it was better at classifying scenarios that it wasn't specifically built or fit from. The way random forests work, means that it takes a bunch of decision trees that don't look at all the data, which makes it a more optimal fit for situations where the data you're classifying isn't exactly the same from the training set. While Decision Tree is massively overfit from training data, random forests is much better. It's training data was 100%, and the test score was 97.05%, the highest of the three for that category.

Figure 1: Task #1 Explanations

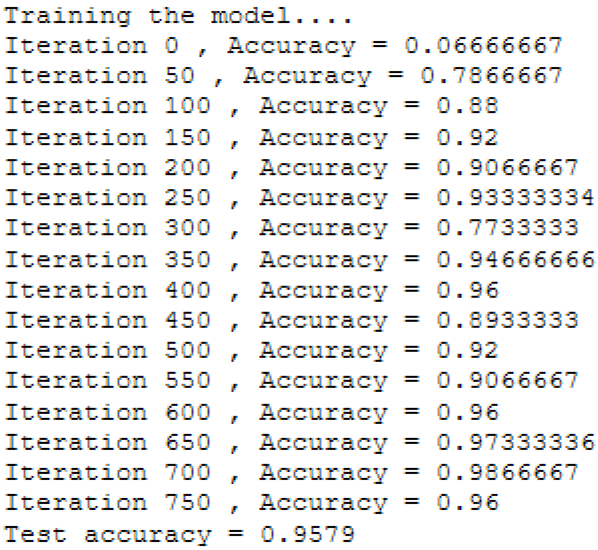
Additionally, I scored the models two different times, on their training data, and their test data. Linear Regression had the lowest drop in accuracy between the training and tests sets, while Random Forests had the highest Testing score. Decision Tree, while have a 100% accuracy on the training set, had the largest drop and lowest test score with 87.55%



	Classifier	Training Score	Test Score
0	Linear Regression	0.933917	0.9255
1	Decision Trees	1.000000	0.8755
2	Random Forests	1.000000	0.9705

Figure 2: Task #1 Classifier Scores

For Task #2, we had to make a convolutional neural network with a 10-class classification with hidden and deep layers. I had 4 assorted layers, 1 being in the output and 2 Fully Connected (FC) layers and 2 Convolutional Layers (Conv). Overall, we ended with an accuracy of 95.79% on the test data. But how do these classifiers work for our own handwriting?



```
Training the model....
Iteration 0 , Accuracy = 0.06666667
Iteration 50 , Accuracy = 0.7866667
Iteration 100 , Accuracy = 0.88
Iteration 150 , Accuracy = 0.92
Iteration 200 , Accuracy = 0.9066667
Iteration 250 , Accuracy = 0.93333334
Iteration 300 , Accuracy = 0.7733333
Iteration 350 , Accuracy = 0.94666666
Iteration 400 , Accuracy = 0.96
Iteration 450 , Accuracy = 0.8933333
Iteration 500 , Accuracy = 0.92
Iteration 550 , Accuracy = 0.9066667
Iteration 600 , Accuracy = 0.96
Iteration 650 , Accuracy = 0.97333336
Iteration 700 , Accuracy = 0.9866667
Iteration 750 , Accuracy = 0.96
Test accuracy = 0.9579
```

Figure 3: CNN Training and Iteration/Testing Accuracies

For Task #3 we first had to draw out 5x10 digits; 5 for each number from 0-9. I did this online and just screen snipped the image, knowing I would resize later down the line. Here are my digits, iterating from 0 to 9 based on row.

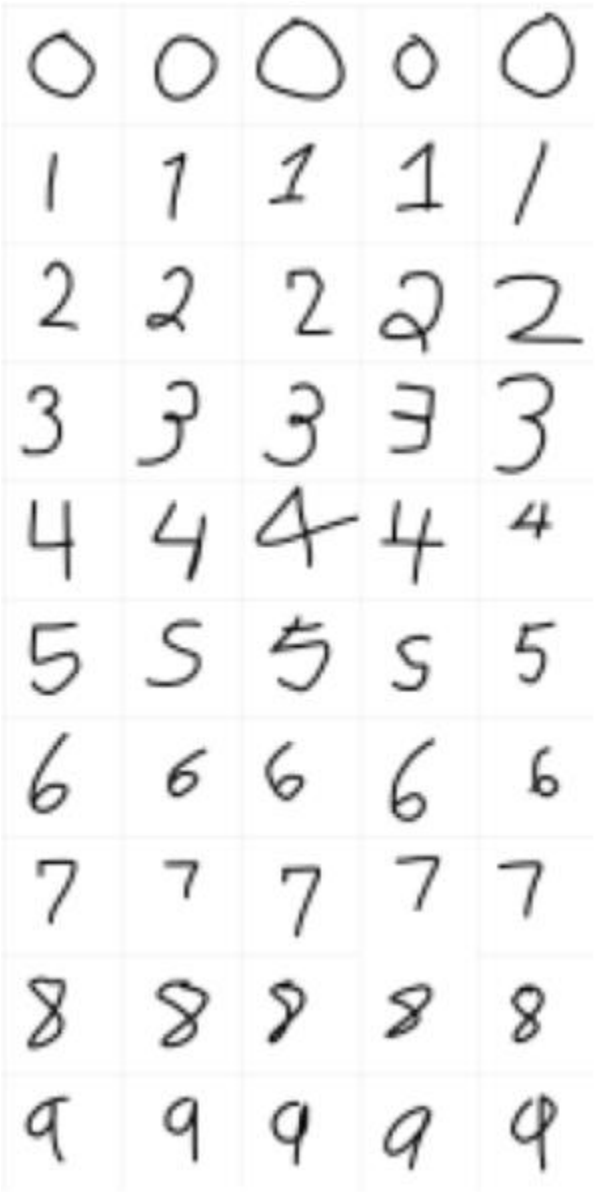
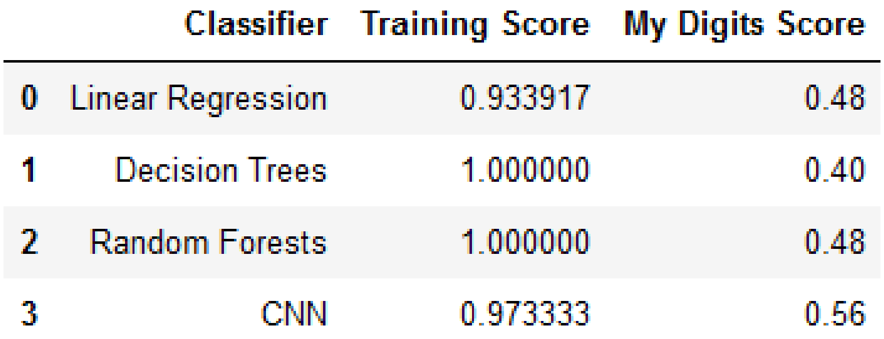


Figure 4: Handwritten Digits

Originally, I thought these digits would slot in quite well with MNIST and would have a decently high accuracy, but I was mistaken. All classifiers had significantly lower accuracies, almost all by half. As I predicted though, the Convolutional Neural Network we put together in task #2 had the highest accuracy, and I figured this because it is the most flexible to new data, which in this case is my own handwritten digits.



	Classifier	Training Score	My Digits Score
0	Linear Regression	0.933917	0.48
1	Decision Trees	1.000000	0.40
2	Random Forests	1.000000	0.48
3	CNN	0.973333	0.56

Figure 5: Classifier Scores for My Digits

Doing my analysis, I can contribute these lower accuracies to a few different reasons. To begin, the images of my own digits aren't the same format, size, and positioned as they are in most MNIST digit cases. For cases like Decision Trees, where they are based so heavily on training data and have little flexibility for new data, this is huge and is what I contribute towards its lower accuracy. CNN had the highest, as previously mentioned I expected, due to its flexibility to new data. Additionally, and I noticed this while screenshotting my data, there is a small amount of border pixels around most of my digit images, most likely from the original screenshot, which could impact the accuracy, although I'm sure not by much. In the end, computers are impressive at predicting under the right circumstances, but in situations where the environment may not be the same (positioning differences, sizing differences and picture borders), some issues and a lower accuracy may be the outcome.