Digit Recognition using Machine Learning

Andy W. Chen[1]

[1]*Sauder School of Business, University of British Columbia, Canada*

DOI: 10.29322/IJSRP.8.6.2018.p7817
http://dx.doi.org/10.29322/IJSRP.8.6.2018.p7817

ABSTRACT: *In this paper, I build automated machine learning models to recognize and predict handwritten digits. The models are supervised learning models used to predict every possible pair of digits. I evaluate the models by comparing the training and testing accuracies to explore the pairs that are most and least difficult to distinguish. I find that the pair 0 and 1 has the highest accuracy and the pair 3 and 5 has the lowest accuracy.*

KEYWORDS: *Digit Recognition, Machine Learning, Logistic Regression, Business Process Improvement Methods*

I. INTRODUCTION

Digit recognition is the process of identifying handwritten digits on objects such as paper, images, and screens of computers and mobile phones using automated systems. Applications of digit recognition are common, especially in improving business processes. For example, a bank may use a digit recognition tool to automatically read the amount on cheques deposited. For another example, post offices may use automated systems to read postal code on letters and packages and route them to the appropriate delivery paths. The task of digit recognition is not trivial. As in natural languages, handwritten digits have endless varieties. Challenges may arise from gathering data, training models, and building robust solutions.

The project's goal is to build automated machine learning models that can recognize handwritten digits with high accuracy. In this project, I build many supervised learning models to predict every possible pair of digits. There are ten digits from zero to nine, giving a total of 45 possible pairs. I compare the training and testing accuracies of every pair of digits to explore which pairs are easiest and most difficult to distinguish. In particular, I build binary logistic regression models, each with two possible outcomes corresponding to one of the two digits being compared.

Related works on digit recognition include one by Duong and Emptoz[1], who proposed a cascade classifier. Their model is a combination of multiple classifiers based on confusion analysis and local retraining mechanism. Garg and Jindal[2] discussed a new feature for recognizing handwritten digits using pixel counting technique and contour following. They found that pixel counting is a very useful method for recognizing ambiguous writing. Azeem et al.[3] attempted to create a model for recognizing Arabic digits and achieved an accuracy of 98.73%. Gattal et al.[4] used a combination of features such as oriented Basic Image Features (oBIFs) and background concavity features in support vector machines (SVM). Aradhya et al.[5] used the nearest neighbor model based on radon transform, which transforms each image into a collection of projections along different directions.

II. METHODS

I build binary logistic regression models to compare every possible pair of digits in the MNIST database. Therefore, I build a total of 45 models. In this database, there are 70,000 images of handwritten images (60,000 are used as training samples and the rest are used as test samples). Each image has a resolution of 28 by 28, giving a total of 784 elements. Each image is stored as a vector of 784 binary numbers where each number indicates the color of the image. The image below shows an example of a handwritten digit in the database. I train each logistic regression model using the training set and predict on the test set the accuracy of classifying each digit for both training and test sets. I then compare the accuracies across the models to explore which pairs of digits are most difficult to distinguish.

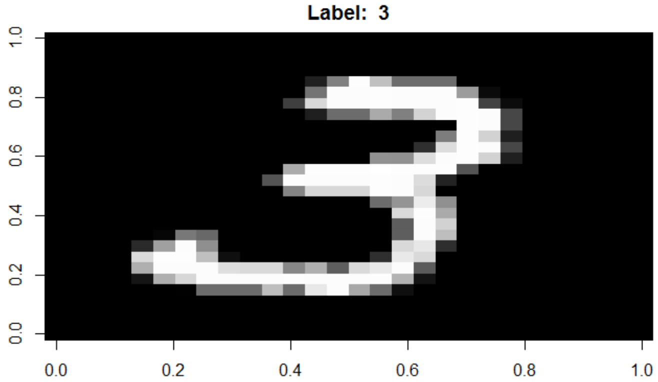

Figure 1. Example of a handwritten digit in the database

III. RESULTS AND DISCUSSION

For each binary logistic regression model, I calculate the accuracy on the training and test set. Table 1 below shows the plot of training and test accuracies for each pair of digits. The results for training data shows that pair 0 and 1 are the easiest to identify, with an accuracy of 99.3%, while pair 3 and 5 are most difficult to distinguish with an accuracy of 80.6%. Figure 2 shows examples of a true positive, true negative, false positive, and false negative for pair 0 and 1. Figure 3 shows examples of a true positive, true negative, false positive, and false negative for pair 3 and 5. For test data, I got the same results. Pair 0 and 1 is the least difficult to identify, with an accuracy of 99.5%, while pair 3 and 5 is most difficult to distinguish with an accuracy of 81.4%. Figure 4 shows examples of a true positive, true negative, false positive, and false negative for pair 0 and 1. Figure 5 shows examples of a true positive, true negative, false positive, and false negative for pair 3 and 5.

Digit 1	Digit 2	Training Accuracy	Test Accuracy
0	1	99.3%	99.5%
0	2	94.7%	94.9%
0	3	91.4%	91.7%
0	4	93.8%	93.9%
0	5	92.6%	93.9%
0	6	94.5%	94.1%
0	7	96.6%	96.0%
0	8	97.4%	97.4%
0	9	98.5%	98.3%
1	2	96.3%	95.3%
1	3	97.9%	98.8%
1	4	97.3%	97.6%
1	5	96.1%	96.6%
1	6	99.1%	99.2%
1	7	96.4%	95.9%
1	8	92.4%	92.8%
1	9	97.9%	98.0%
2	3	92.8%	92.9%
2	4	93.8%	93.3%
2	5	94.2%	95.1%
2	6	95.6%	95.7%
2	7	95.9%	95.8%
2	8	90.0%	90.2%
2	9	96.9%	96.7%
3	4	97.3%	98.3%
3	5	80.6%	81.4%
3	6	96.6%	97.0%
3	7	95.6%	95.6%
3	8	92.9%	93.5%
3	9	96.7%	96.8%

4	5	97.3%	97.3%
4	6	91.4%	91.4%
4	7	96.5%	96.8%
4	8	94.8%	94.8%
4	9	87.6%	87.6%
5	6	91.5%	91.6%
5	7	96.9%	96.2%
5	8	89.6%	88.8%
5	9	93.8%	94.4%
6	7	99.0%	99.2%
6	8	93.8%	94.4%
6	9	95.1%	94.1%
7	8	96.3%	95.9%
7	9	90.4%	90.6%
8	9	93.4%	93.0%

Table 1. Training and test accuracies for each pair of digits

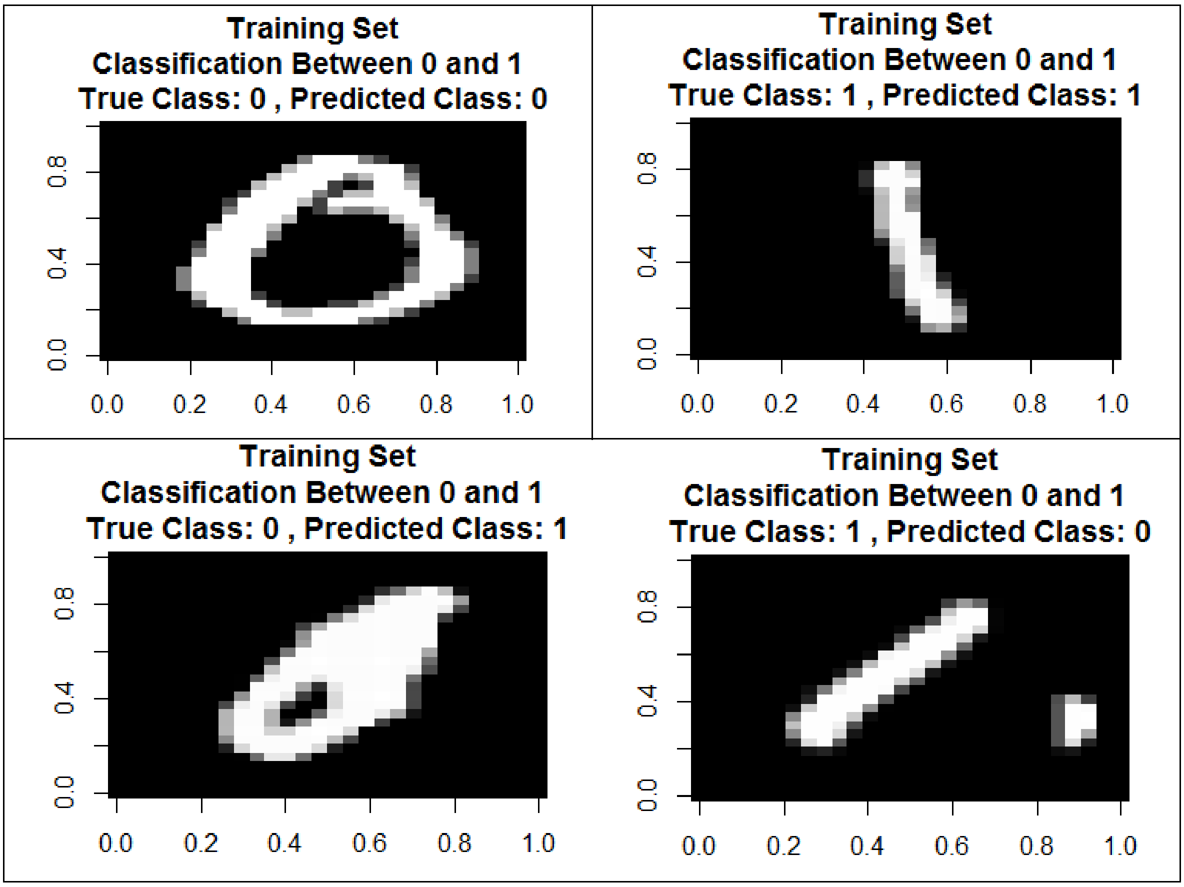

Figure 2. True positive, true negative, false positive, and false negative predictions for 0 and 1 (Training Set).

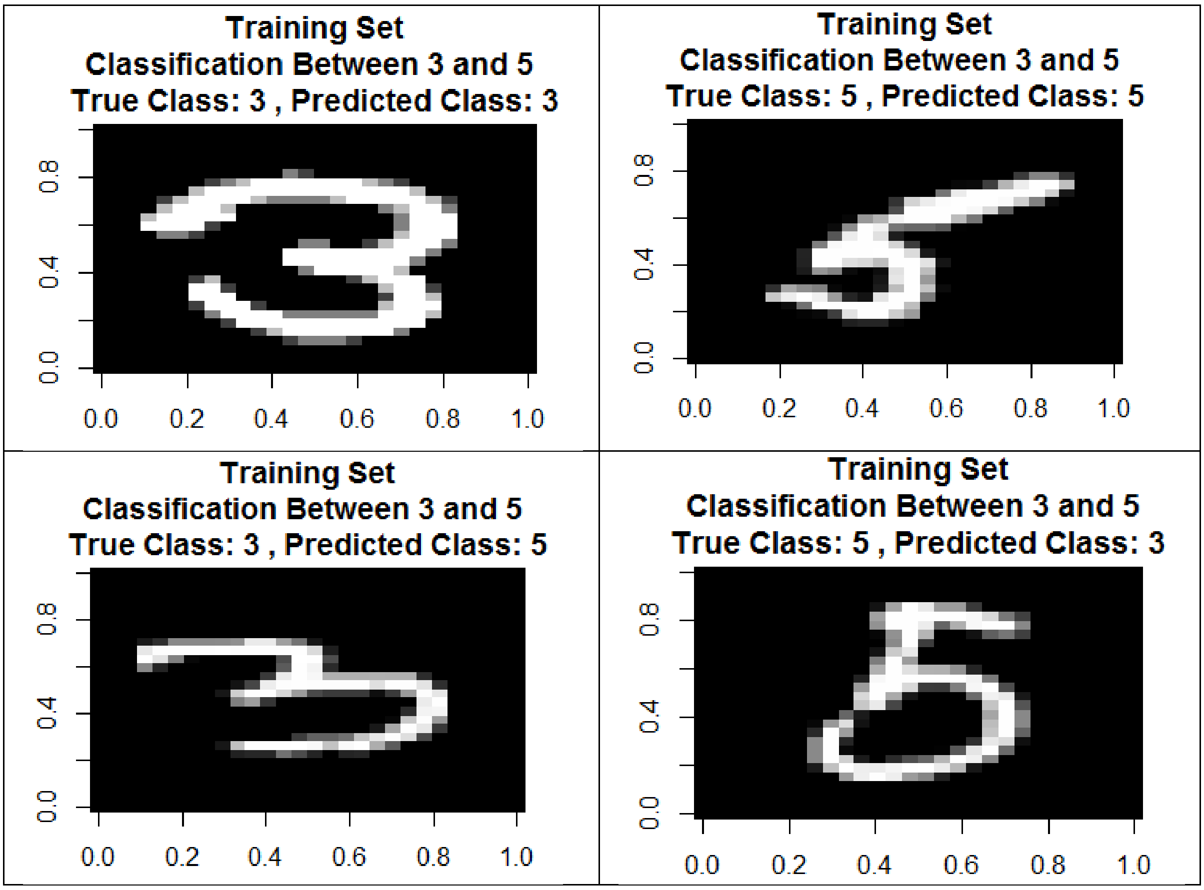

Figure 3. True positive, true negative, false positive, and false negative predictions for 3 and 5 (Training Set).

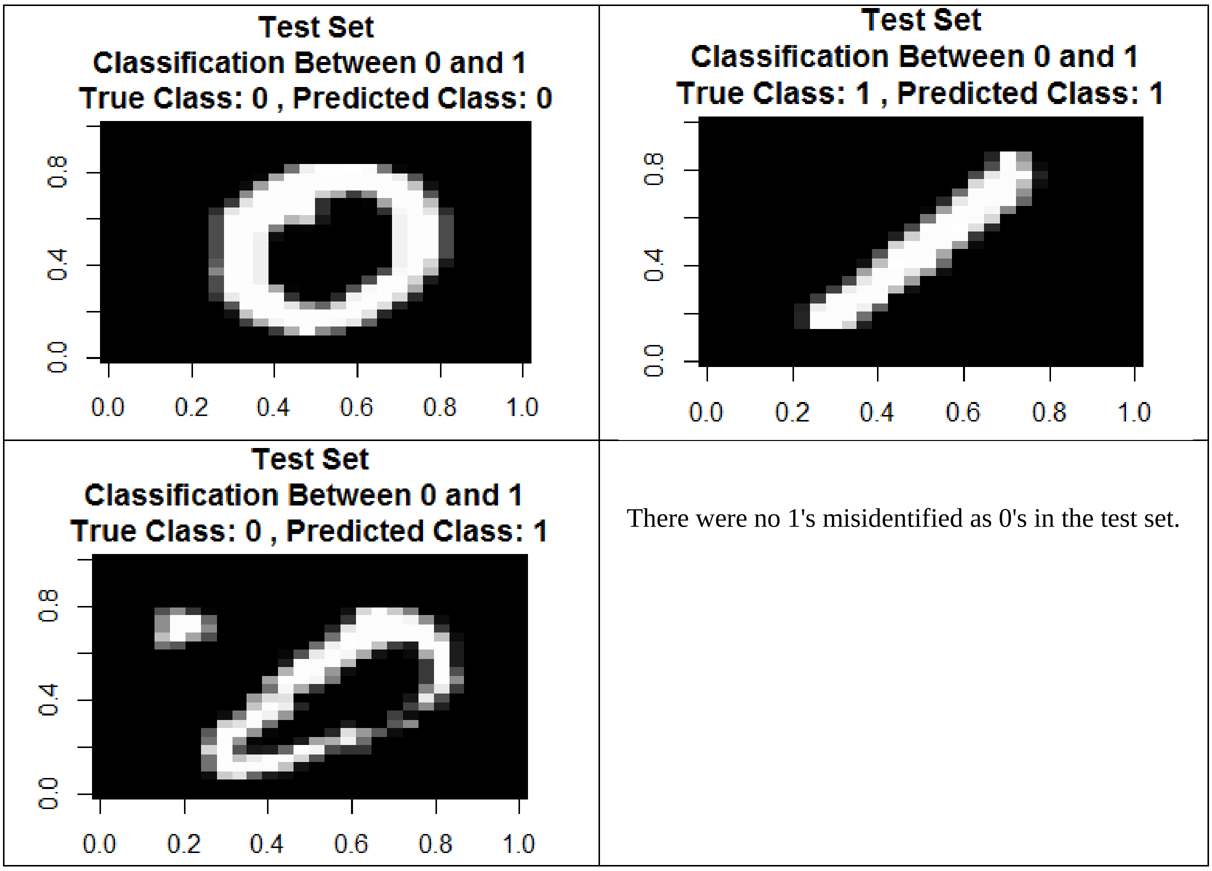

Figure 4. True positive, true negative, false positive, and false negative predictions for 0 and 1 (Test Set).

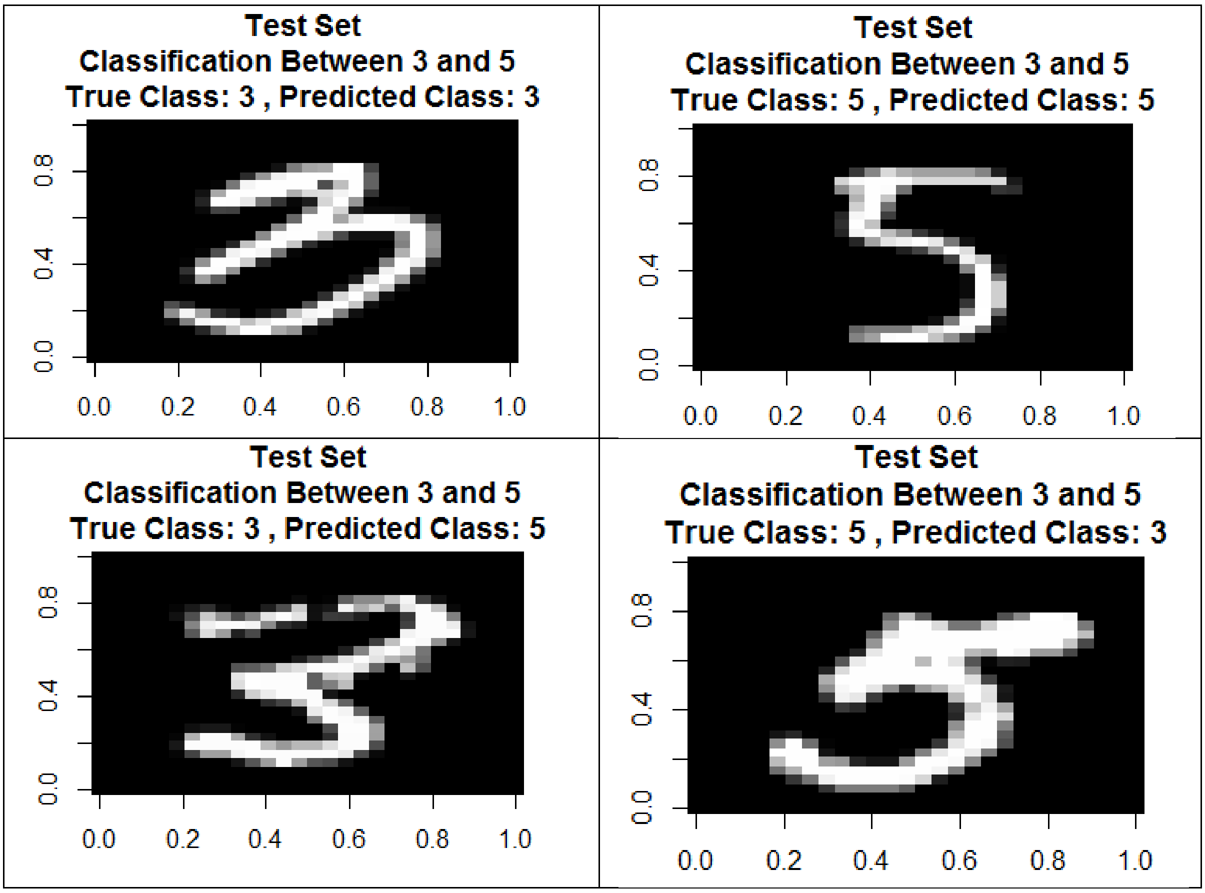

Figure 5. True positive, true negative, false positive, and false negative predictions for 3 and 5 (Test Set).

The results show that pair 0 and 1 has the highest training and test accuracies, while pair 3 and 5 has the lowest training and test accuracies. This is reasonable because 0 and 1 look quite distinctive from each other, so do digits 3 and 5. In other words, it is very unlikely that someone would write 0 that looks like a 1 and vice versa. On the contrary, there many examples where a 3 and 5 look similar. Both digits contain curvy parts and people with different habit and possibly different cultures can write these two digits in ways hard to distinguish.

IV. CONCLUSION

This paper shows the results of pairwise comparison between handwritten digits in terms of prediction accuracies. Digits that look distinctive from each other are reasonably easy to distinguish. However, some pairs of digits that look similar will test the limits of machine learning models. As a future extension, more comprehensive training data may be used so the models can be trained on a more diverse range of handwritten digits. Other supervised learning models such as support vector machines and decision trees may also be used.

REFERENCES

[1] Duong J, Emptoz H. Cascade classifier: design and application to digit recognition. Eighth International Conference on Document Analysis and Recognition (ICDAR'05). 2005;2:1065-1090.

[2] Garg NK, Jindal S. An efficient feature set for handwritten digit recognition. 15th International Conference on Advanced Computing and Communications (ADCOM 2007). 2007;540-545.

[3] Azeem SA, Meseery ME, Ahmed H. Online Arabic handwritten digit recognition. 2012 International Conference on Frontiers in Handwriting Recognition. 2012;135-140.

[4] Gattal A, Djeddi C, Chibani Y, Siddiqi I. Isolated handwritten digit recognition using oBIFs and background features. 2016 12th IAPR Workshop on Document Analysis Systems (DAS). 2016;305-310.

[5] Aradhya VNM, Kumar GH, Noushath S. Robust unconstrained handwritten digit recognition using radon transform. 2007 International Conference on Signal Processing, Communications and Networking. 2007;626-629.