Handwritten Digit Classification

Author: SM

Abstract—In this paper, we evaluate the behavior and performance of 3 supervised classification algorithms including a linear regression learner, a logistic regression learner, and a k-nearest neighbor classification learner. We implement the codes in R, and we use the accuracy, misclassification rate, and time-to-train as evaluation metrics.

1 INTRODUCTION

The idea is to implement a classification learner to classify the handwritten digits. This learner can be part of an online order processing systems in healthcare, insurance, banking, postal service or even library. The system accepts a scanned image of a prescription, repot, cheque, address, etc.(Anil C.,2022). Then, the image processing techniques are applied to adjust the contrast, sharpness, and brightness. Next, feature extraction is performed to identify the key information and generate the vector representation of the characters. These vectors are fed to a classification learner to identify the contents. The variety comes from different handwriting styles, and poor light conditions while scanning the documents.

In this article, we focus on a small subset of this problem: handwritten digit classification.

Problem Statement: The objective of our analysis is to classify the handwritten digits between 2 and 7.

Data Set: The dataset provided includes normalized handwritten digits, scanned from envelopes by the U.S. Postal Service. Pre-processing and feature extraction techniques are already applied on the original scanned images with different sizes and orientations (Le Cun et al., 1990). Each image is represented by a label-id 0-9 (V1), and a vector of 256 grayscale values (V2-V257). The detailed description can be found from Stanford's website. We filter the rows with V1 equal to either 2 or 7 and save it to ziptrain27, and ziptest27 for further investigation.

2 EXPLORATORY DATA ANALYSIS

We start our analysis by perfuming initial investigations on data to understand the features better and to discover any patterns or anomalies in the data set. We use summary statistics and visual representations to investigate the variation of features.

Size of dataset: Training data includes 1376 images each containing 256 features. 731 images are labeled as 2, 645 images are labeled as 7. Test data includes 345 images. 198 images are labeled as 2, 147 images are labeled as 7. Target value distribution between training and test data is maintained.

Data set	Distribution of 2	Distribution of 7
Training data	53%	47%
Test data	57%	43%

Table 1: Distribution of target values in training and test data set

Variable descriptions: Each scanned image is normalized to a 16 x 16 grayscale images. Each image is represented by vector of 256(= 16 x 16) grayscale values.

Range: All the 256 features are in [-1, +1]. There is no need for standardization.

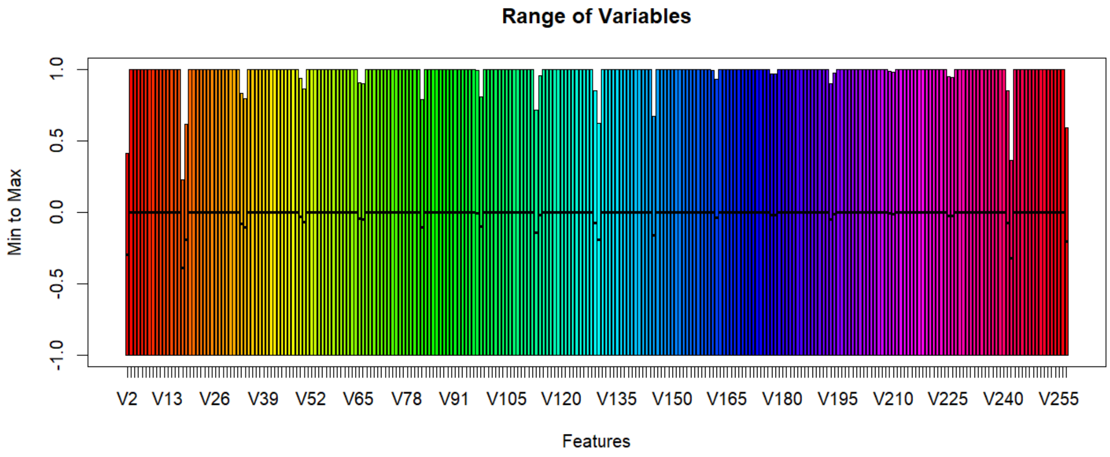


Figure 1: Range of Variables; We could see all features are within the [-1,1].

Correlation: Next, we investigate the correlation between each pair of features. If there is a strong correlation between two features, we keep only one. In other words, we reduce the complexity of the model by dimensionality reduction

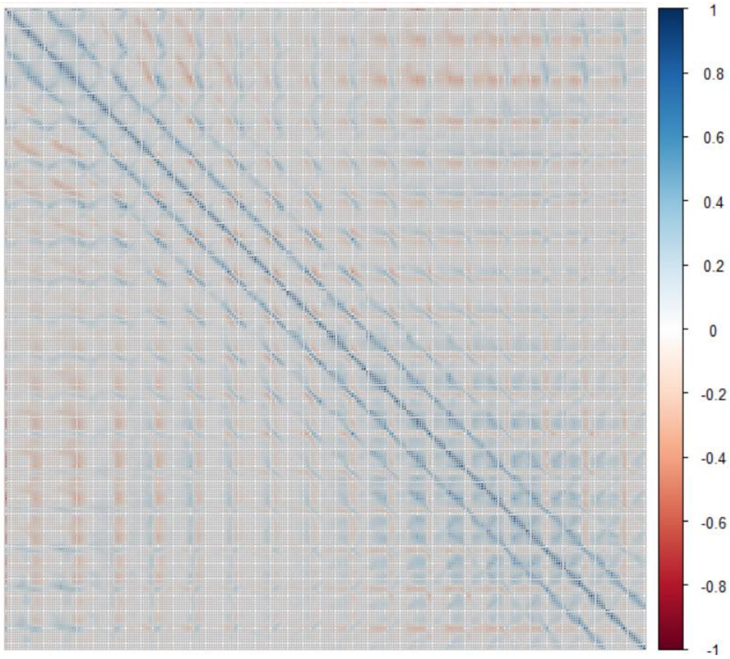


Figure 2: Correlation Heat map between each feature. Labels are removed for clarity.

We can observe strong correlation between adjutant features. We are working with image data, and many adjacent pixels have same grayscale value for the same digit. We remove highly correlated features (cutoff = 0.7) and end up with only 75 features. Figure 3 illustrate which parts of a 16 x 16 image are not correlated and would be used to classify between 2 and 7.

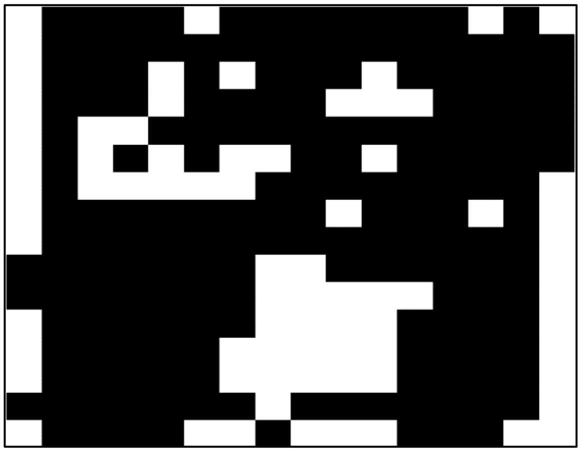


Figure 3: An illustration of selected features (white) in a 16 x 16 image

Outliers: Next, we investigate the outliers using boxplot, but decided not to remove the outliers and just work with the data set as it is. Because of the variety in handwriting style, digits may be skew towards right, left, top or bottom. In other words, for some features we may have rare values- but it is still expected.

Variation: Next, we explore the variation of each predictor for different response. The goal is to develop an **intuitive** understanding about the data. From figure 4, we can **anticipate** that V17, V113, V129 would be more useful than V7 while distinguishing between 2 and 7.

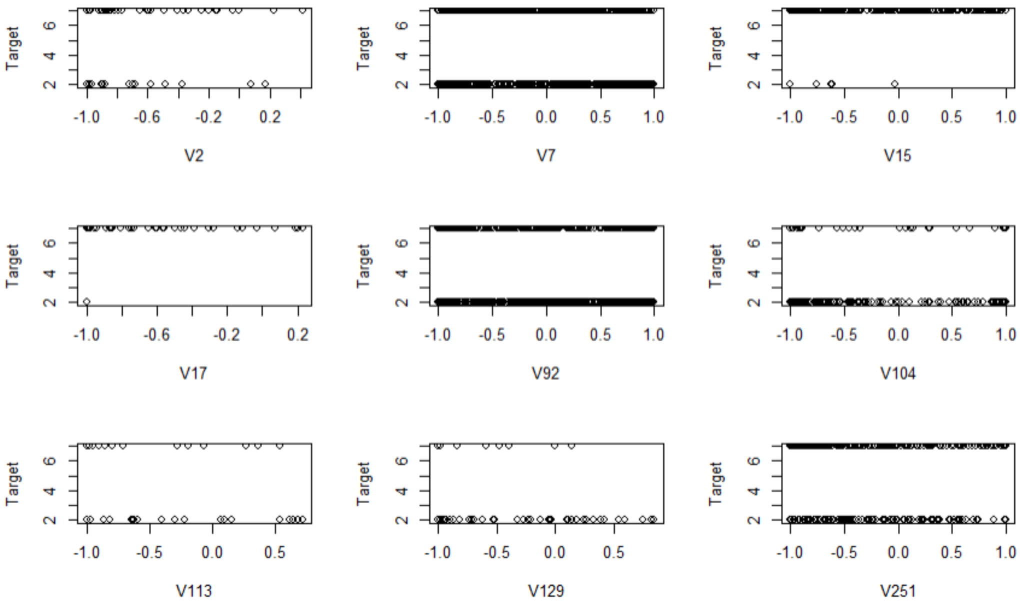


Figure 4: Sample plot of variation of response vs. each feature. Note that , for target value of 2, V17 seems to be -1, and V15 seems to have limited values.

3 METHODOLOGY

In this section, we discuss the classification algorithms. We create linear regression, logistic regression, and KNN classifier. We understand that linear regression is not the right method for *binary classification*. We simply perform linear regression just because it is asked in the homework questions. Because the linear regression output continues numbers out of desired range, we use a transformer function *2 + 5*(Linear. Regression >= 4.5)* to generate the binomial output.

First, we trained linear regression, logistic regression, and KNN classifier using all the training data and calculated the error rate for training data(ziptrain27), and test data(ziptest27). Error rate refers to misclassification rate from the confusion matrix.

Method	**Training Error rate**	**Testing Error rate**
Linear Regression	0.01962209	0.04347826
Logistic Regression (p=0.5)	0.007994186	0.04057971
KNN (k=3)	0.009447674	0.01739130

Table 2: Error rate for each method without using cross-validation.

For linear regression, we should not refer to the R^2 as a measure of quality, because R^2 from lm is calculated before applying the binomial-transformation.

Overfitting: Figure 5 illustrate the overfitting for KNN. When K is reduced from 3 to 1, the training error rate decreases but the testing error increases. This is because the model overfit to the training data and captured noise.

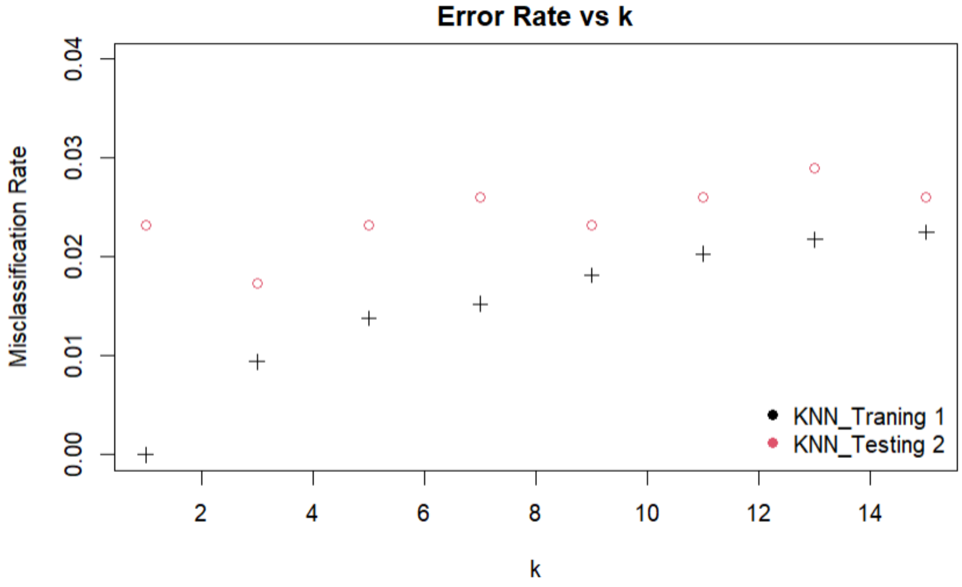
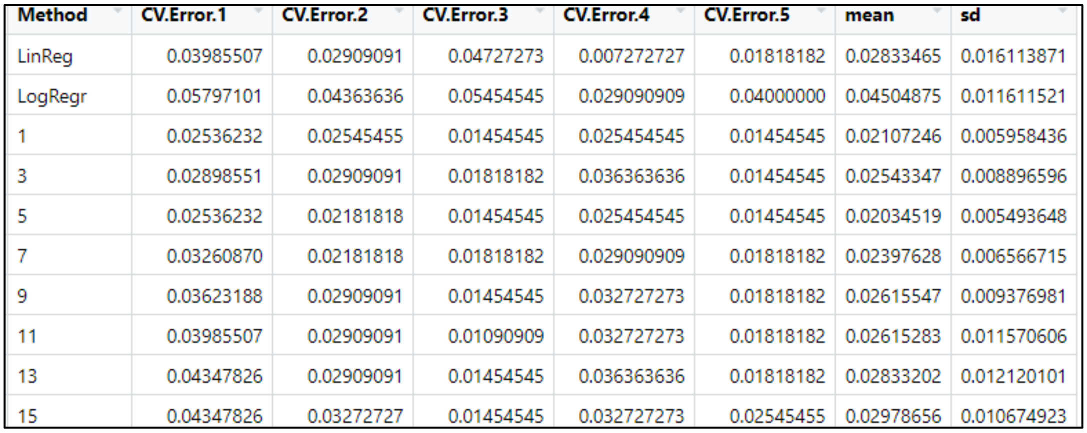


Figure 5: Overfitting in KNN when K = 1.

Based on Table 2, and Figure 5, and without performing any cross-validation, we can say Logistic Regression performs better on training set, and KNN with K=3 performs better overall.

Cross-Validation: We utilize k-fold cross validation with k=5. Our approach is a bit different from what is suggested in homework questions: a) We divide the training dataset(ziptrain27) into 5 subsets. b) At each step, we train all the 3 models on all subsets but one. c) Then, we calculate the misclassification rate for all 3 models using the remaining subset. d) We use mean/variances of the testing errors to compare these models and pick the best one. e) Finally, we report the performance of our chosen model by testing it on test data set (ziptest27).

At each step of the cross-validation, we calculate the error rate for linear regression, logistic regression with different probability cutoff values, and KNN with different K values. Results are listed in table 3. We can see that KNN with K=5 has the lowest average misclassification error, and standard deviation. K=3 also has a low average and standard deviation error rate.



Method	CV.Error.1	CV.Error.2	CV.Error.3	CV.Error.4	CV.Error.5	mean	sd
LinReg	0.03985507	0.02909091	0.04727273	0.007272727	0.01818182	0.02833465	0.016113871
LogRegr	0.05797101	0.04363636	0.05454545	0.029090909	0.04000000	0.04504875	0.011611521
1	0.02536232	0.02545455	0.01454545	0.025454545	0.01454545	0.02107246	0.005958436
3	0.02898551	0.02909091	0.01818182	0.036363636	0.01454545	0.02543347	0.008896596
5	0.02536232	0.02181818	0.01454545	0.025454545	0.01454545	0.02034519	0.005493648
7	0.03260870	0.02181818	0.01818182	0.029090909	0.01818182	0.02397628	0.006566715
9	0.03623188	0.02909091	0.01454545	0.032727273	0.01818182	0.02615547	0.009376981
11	0.03985507	0.02909091	0.01090909	0.032727273	0.01818182	0.02615283	0.011570606
13	0.04347826	0.02909091	0.01454545	0.036363636	0.01818182	0.02833202	0.012120101
15	0.04347826	0.03272727	0.01454545	0.032727273	0.02545455	0.02978656	0.010674923

Table 2: Comparison of different classifiers using 5-fold cross-validation.

Based on Table 2, we pick the KNN with K=5 as the best model. Next, we build a KNN classifier using the complete training set (ziptrain27), and we report the estimated performance based on model prediction on test set(ziptest27): 2.03%

4 SUMMARY AND CONCLUSION

We analyzed the behavior and performance of 3 classification learners. We learned that small K may cause overfitting in KNN algorithm. We also learned that comparison of different algorithm with or without cross-validation may lead to different decision. It is always better to utilize cross validation when comparing different algorithm.

It was a surprise that the logistic regression with optimized cutoff value underperformed the linear regression. Our first assumption was that linear regression shall not be used for binomial classification, but the results are quite impressive.

And finally, we find out that KNN with K=5 is the best model to classify the handwritten digits between 2 and 7 . We report the estimated performance based on model prediction on ziptest27 as misclassification rate = 2.03%.

5 REFERENCES

1. Lecun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In D. Touretzky (Ed.), Advances in Neural Information Processing Systems (NIPS 1989), Denver, CO (Vol. 2). Morgan Kaufmann.
2. Anil C.(2022). How to easily do Handwriting Recognition using Machine Learning. Nanonets

6 APPENDIX

Codes are developed in R and attached in the R Markdown file.