Nicholas Lichtsinn

IST 707

Homework 6/7

**Introduction**

Handwriting analysis has many uses in modern society from digitizing historical documents to recognizing signatures for point-of-sale systems and being able to deposit checks on a mobile app. Each of these methods are used daily in our society with no thought as to how it processes on the back end.

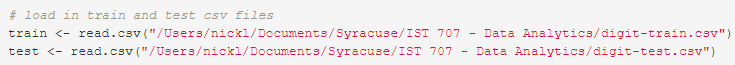
In 1999 this dataset of handwritten images has been used to test and train classification algorithms and models. The Modified National Institute or Standards and Technology or more commonly as MNIST, has maintained a database with thousands of training and testing images to test these models. This paper will look at a small subsection of this database to create various models for this classification task.

This paper will use Decision Trees, Naïve Bayes, k – Nearest Neighbors (kNN), Support Vector Machines (SVM), and Random Forest models to classify printed numeric characters. Comparing each method’s accuracy.

**Analysis**

Data Preparation:

Two csv files were used from Kaggle.com for this research. The train.csv has 42,000 observations with 784 variables and the test.csv has 28,000 observations with 784 variables.

Both files were loaded into R with read.csv.

Chart, histogram

Description automatically generatedA distribution of the data was created to visualize the values of the training set

Text

Description automatically generatedSamples were then created from the train and test datasets due to their large size.

Next all samples with zero in all images and samples with low variance were removed and a clean dataset was created with the remaining observations. A min/max function was used to normalize the data and a sample test and train set were created from the clean data, using 80% of the values for train data and 20% for test.

**Results**

**Diagram

Description automatically generatedTable

Description automatically generated**Once the dataset was cleaned and train and test sets were ready the classification models could be created. First a decision tree model was created using rpart and a Root node error: 2968/3360 = 0.88333 was calculated. The decision tree model had an accuracy of 1 and there are definite signs of overfitting.

Table

Description automatically generatedThe Naïve Bayes model was created and tuned with kernel and with laplace smoothing effects, neither of which had any effect and the same Accuracy of 70.12% was found.

To create the kNN model, the train and test samples were separated into labels and numbered sets to train the model. Both Euclidean and cosine distance measures were tested, and the Euclidean resulting accuracy score was 82.86%.

Table

Description automatically generated

Table

Description automatically generatedFor the SVM model linear, radial and polynomial kernels were used, with the radial kernel giving the highest accuracy score of 91.55% (Linear: 88.93%, Polynomial: 88.45%). This makes sense as the radial kernel is a great option for non-linear data and when there is no prior knowledge of the data for the model.

Last a Random Forest model was created using 30, 100 and 200 trees. With 200 trees being the highest at 90.95% (30: 86.9%, 100: 90.12%). The following graph shows the error decreasing with the number of trees used in the model.

Table

Description automatically generated

Histogram

Description automatically generated

After reviewing each model and their highest performance the SVM model led the way at 91.55%, then Random Forest at 90.95%, kNN at 82.86% and the Naïve Bayes model at 70.12%.

**Conclusion**

There is no golden model for every classification problem and there are many ways to tune each model depending on how the data is processed and what classification is being attempted. For this dataset the SVM and Random Forest performed higher than the kNN model and much higher than the Naïve Bayes model after tuning and there was an issue with overfitting for the Decision Tree Model.

Going forward there are some additional steps that could be taken to increase the accuracy of these models with more tuning or different ways of processing the data could be introduced. This is a problem that will continue to be researched as new technologies become available and new ways of implementing handwriting analysis are needed.