Kellen Tsuruoka

[ktsuruok@syr.edu](mailto:ktsuruok@syr.edu)

Homework 6

Handwriting Recognition

Naïve Bayes vs Decision Trees

August 17, 2022

Naïve Bayes vs Decision Trees for Handwriting Recognition

Computers have come a long way since their invention in the 1940s. The computer's ability to process information has improved exponentially since its inception. Modern-day supercomputers have almost reached the computation level of the human brain. Pattern recognition is an area where modern-day computer programs are catching up to humans. Computers still have trouble picking up subtleties in images.

Machine learning algorithms help computers mimic the abilities of the human brain. The Modified National Institute of Standards and Technology (MNIST) is the standard for testing machine learning techniques. MNIST is a set of handwritten images that test the machine learning algorithm's ability to classify the images.

The remainder of this paper will use the MNIST data set to compare two different machine learning techniques. The goal is to juxtapose the two techniques to see their strengths and weaknesses. The overall benchmark is how accurately the algorithms can decipher handwritten images.

**Analysis:**

The MNIST data set contains handwritten numbers recorded in the pixels. The pixels are recorded by measuring how much the image crosses over the individual pixels. A good analogy is writing numbers on graph paper with a thick marker. The boxes on the graph paper are assigned a number. The density marker is recorded as pixel data. The MNIST data set is similar but records the brightness in greyscale.

In the dataset, the pixel numbers are the column values. There are seven hundred- and eighty-five-pixel columns in the data set. The first column, labels, in the data set specified the handwritten number. Each row contains the pixels for one number. Figure 1 is an example from the data's CSV file showing the structure of the data set. The data values are not binary. Figure 2 shows the values that correspond to brightness in greyscale.



Figure : MINST Data Set Example



Figure : MINST Example with Values

The data required only a few modifications to run the analysis. There were no missing or NA values in the data set. The label attribute was transformed into factorized data to run correctly. Supervised machine learning algorithms require nominal data to be factorized to run correctly. If the nominal attributes are not factorized, the algorithm would not know that the values are different. The data was divided into a training and testing set. There were already training and testing datasets available to download from Kaggle. However, in the interest of time and computing power, only the training data file was used in this analysis. The training data set was already around 77mbs. The file was separated into a training and testing data set. The training set was about two-thirds of the file, and the testing data was the remaining third. The labels were removed from the testing set so the models would run properly.

Figure 3 is a bar graph of the frequency of the numbers in the dataset. There is an even distribution of numbers in the dataset. The training data needed an even distribution of numbers to give the model the best chance of predicting the digits. Figure 4 shows the distribution of the numbers of the training data. Luckily, the training data sample is a good representation of the complete data set.



Figure : Count of Numbers in the Dataset



Figure : Training Data Digit Distribution

The Naïve Bayes classification model is one of the models used to predict handwritten digits. The Naïve Bayes model uses probabilities to predict outcomes based on the Bayesian statistics method. In this case, the model uses the partitioned training data to calculate conditional and posterior probabilities for the handwritten numbers using the greyscale pixels. The model predicts the handwritten digits by calculating the probability of a number from the value of the pixels.

Decision Trees are the second model used in this analysis. Decision trees work by creating partitions in the data and creating branches from the partitions. For example, a binomial variable is a natural partition in the data that would create a branch in the decision tree. The decision tree predicts the handwritten digits by creating partitions in the pixelated data for every number. The unlabeled testing data is partitioned based on the partitions established from the labeled training data.

**Technical Analysis:**

Two methods of Naïve Bayes algorithms were tested to see if they yielded different results. The first model runs using the e1071 package. The model ran with standard parameters. Overall, the model predicted the training data with 52% accuracy. The model predicted 0, 1, 6, and 9 well. The strong results could be due to the numbers having distinct shapes. For example, the model did not perform well predicting 4 and 7. The model classified the majority 4's and 7's as 9's. Which makes sense considering the 4 could be interpreted as a 9 missing a curly tail. The 7's can be considered 9's that are not closed on top.

Calendar

Description automatically generated

Figure : e1071 Naive Bayes Model Confusion Matrix



Figure : e1071 Model Prediction Distribution

The naivebayes package created the second Naïve Bayes model. A second model was created to test if another package would yield different results. The naivebayes package produced the same accuracy as the e1071 package.

Calendar

Description automatically generated

Figure : Naive Bayes Model 2 Confusion Matrix



Figure : Naive Bayes Model 2 Plot

The third Naïve Bayes model ran with the use of the usekernel parameter. The usekernel parameter is a smoothing technique using kernel density estimation(KDE) to improve the accuracy of Naïve Bayes models. The KDE creates a smooth line of best fit around the data points. The KDE is the calculation of the weighted distances between the data points and the line of best fit. The KDE model provided the worst accuracy compared to the other Naïve Bayes models.

Calendar

Description automatically generated

Figure : Naive Bayes Model 2 Confusion Matrix

The best performing Naïve Bayes model was created by adding the usepoisson parameter. The usepoisson parameter uses the Poisson distribution to estimate the conditionals for non-negative integer values. Poisson distribution is typically used when working with time series data. In this case, perhaps the Poisson distribution helped to establish a better training model by digits as individual events to create more substantial probabilities. The model produced an overall accuracy of 76%, as seen in figure 10. The model predicted numbers 2 – 5, 7, and 8 better than the previous Naïve Bayes models. Interestingly, the model performed worse predicting 0 – 1 and 9 than the other models.

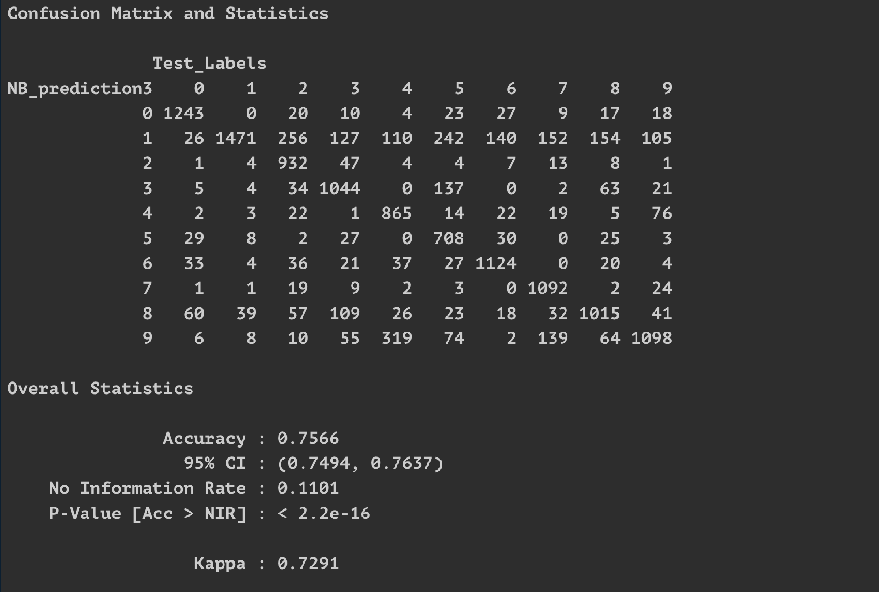


Figure : Naive Bayes Model Three Confusion Matrix



Figure : Naïve Bayes Model Three Plot

Two decision tree models were created to compare to the Naïve Bayes model. The first model ran with no additional parameters. Compared to the standard Naïve Bayes model (figure 5), the decision tree performed better, with an overall accuracy of 60% (figure 12). The decision tree created is seen in figure 13.

Calendar

Description automatically generated

Figure : Decision Tree One Confusion Matrix



Figure : Decision Tree One

The final decision tree ran using rpart.control parameters. The additional parameters added were minsplit = 5, maxdepth = 3, and cp = 0.01. The goal of the added parameters was to create a tighter model to increase accuracy. Unfortunately, the added parameters yielded the opposite results. As seen in figure 14, the accuracy was 48%. The model had 0% accuracy for 6 and 9. The restricted parameters must have made it difficult for the model to decipher between numbers that had similar shapes.

Calendar

Description automatically generated

Figure : Decision Tree Two Confusion Matrix



Figure : Decision Tree Two

For digit predicting using the MNIST dataset, the Naïve Bayes model with Poisson smoothing had the best accuracy at 76%. The decision tree without additional parameters correctly predicted the training data at a 60% success rate. Naïve Bayes and Decision Tree models have their strengths and weaknesses in data models. In this case, Naïve Bayes outperformed Decision Trees in predicting handwritten numbers from the NMIST dataset.

**Conclusion:**

The Modified National Institute of Standards and Technology (MNIST) is the standard for testing machine learning techniques. MNIST is a set of handwritten images that test a machine learning algorithm's ability to classify the images. It is critical to have a standard dataset to test new machine learning algorithms that are developed.

MNIST can also compare established machine learning algorithms to compare performance. It is an opportunity for learners to experiment with different machine learning techniques. Learners can use the experience when working with data science problems later.

The two algorithms compared in this paper both have the same end goal. However, from a learner's perspective, assessing when it is optimal to use each model and what datasets are best suited is helpful. Part of the learning process is using trial and error to develop a knowledge base.