Rohini Shrivastava

HW 6

**INTRODUCTION:**

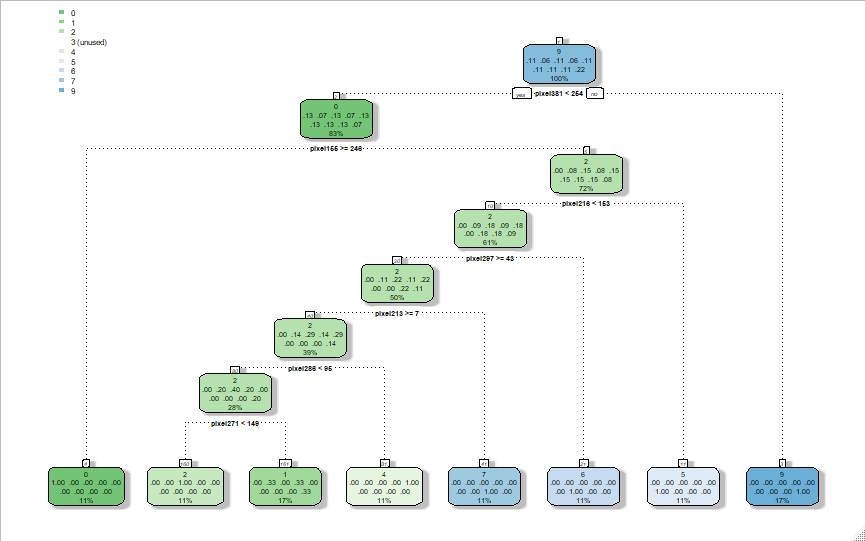
The MNIST (Modified National Institute of Standards and Technology) dataset is a large database of digits that are handwritten. It is commonly used for training various image processing systems. The MNIST dataset is also used for training and testing in the field of machine learning. It was created by "re-mixing" the samples from NIST's original datasets. It contains 60,000 small 28x28 pixel images. The digits are between 0 and 9.

The data was downloaded through Kaggle. In the download, we received a training set and a testing set. The training set had 785 columns, with the first column called label. The test set is the same as the training set, but it does not contain the “label” column. Both the test and training set were imported into R and a data frame was created for both. This data frame was then used to run Decision Trees and Naïve Bayes models.

**DECISION TREE:**

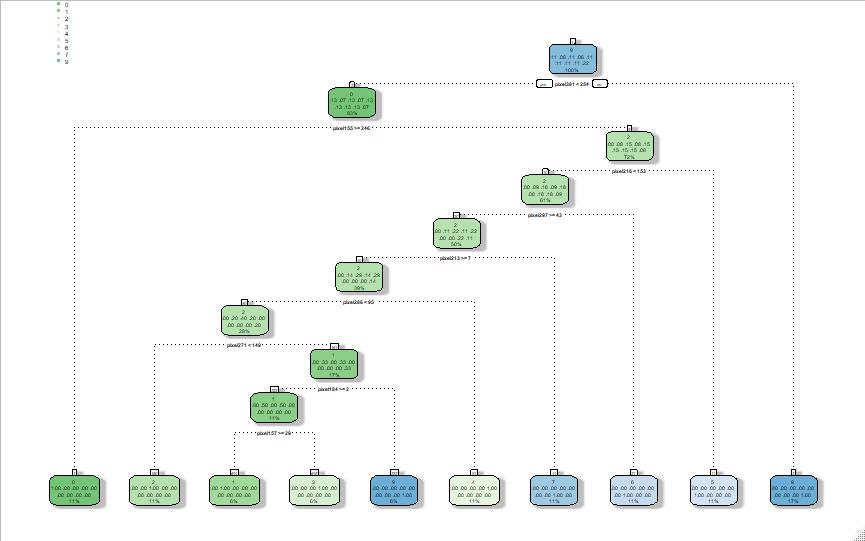
A decision tree is a diagram that is used to determine an action or show a probability. It forms a tree-like structure with each branch showing a possible decision, outcome, or reaction. Three decision tree models were run. Each model used the *rpart* function and the training data set. The model was then used to predict the test dataset.

The first model that was run, there was minsplit of 5 and a maxdepth of 10. In this situation there needs to be a minimum of five observations in the node before a split is done, and the tree will go ten nodes deep maximum (*Figure 1).* Most of the data set showed either a 0 or a 9 for the handwriting in the decision tree model. When a prediction was ran and compared to the test set with the labels, we found it to be 77.77% accurate.



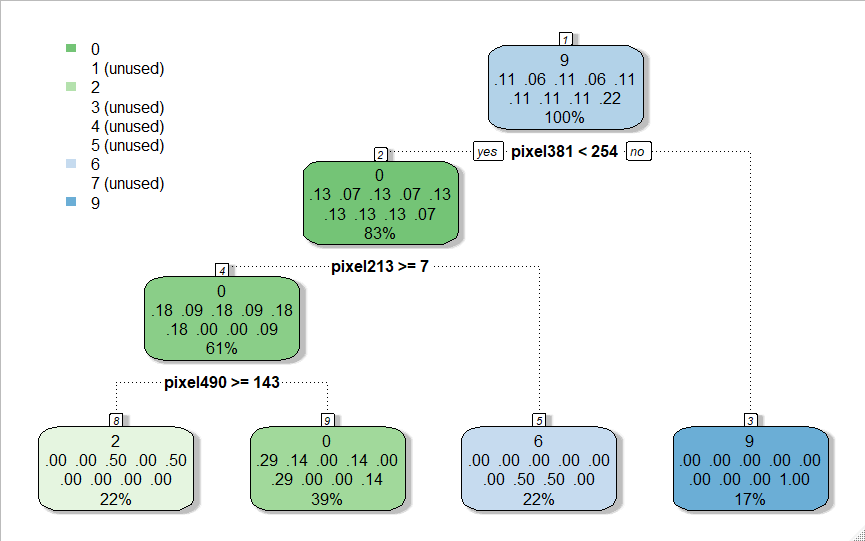
*Figure 1: First Decision Tree*

The first model that was run, there was minsplit of 2 and a maxdepth of 13. (*Figure 1).*  When a prediction was run and compared to the test set with the labels, we found it to be 71.46% accurate.



*Figure 2: Second Decision Tree*

The last decision tree was a model with a minsplit 8 of and a maxdepth of 5 (*Figure 3).* It was shown to be 81% accurate.

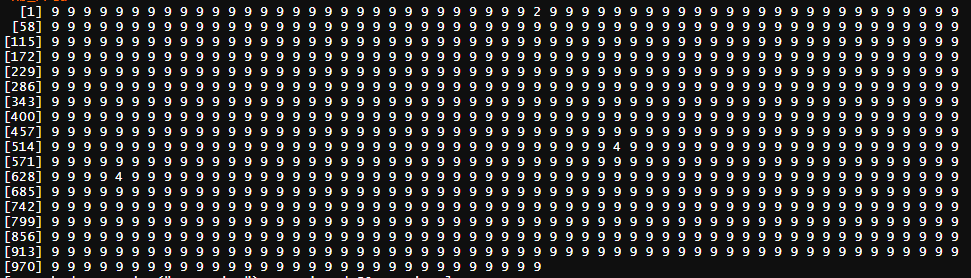


*Figure 3: Third Decision Tree*

When comparing the three models, the last decision tree that was run was the most accurate at 81% this had the least amount of depth, yet the highest amount of minsplit.

**NAÏVE BAYES:**

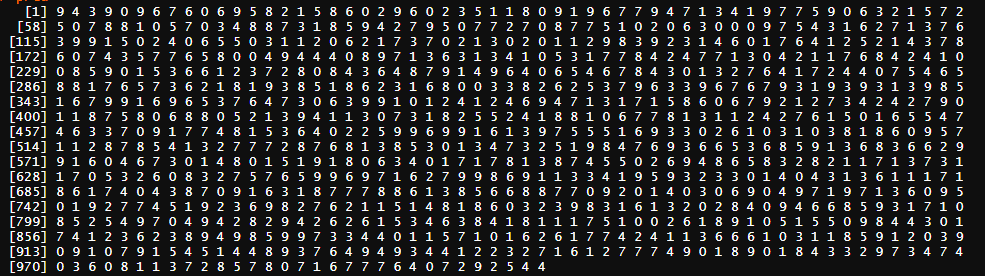
Naive Bayes is a Supervised classification algorithm. it makes the assumption that the occurrence of a feature is independent from the other features. Similar to the decision tree, a model was created using the train data set. The labels were removed and run for a prediction with the test set. When the Naïve Bayes model ran, most of the predictions were ‘9’ (*Figure 4).* This seems highly unlikely. When trying to run a confusion matrix, there were a series of errors due to the size of the data sets. However, looking at this set, we can assume that the accuracy is less than 50%.



*Figure 4: Naïve Bayes Model Prediction*

**KNN:**

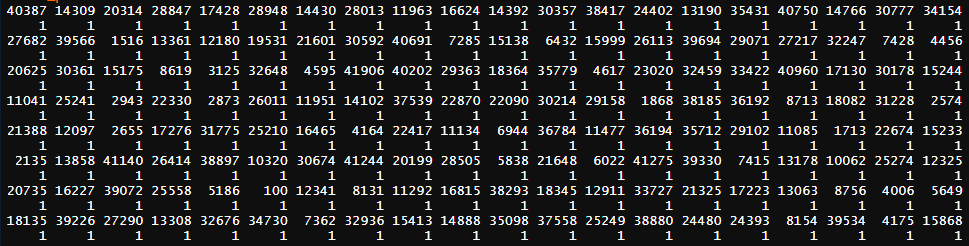
K-Nearest Neighbor is a model that classifies data points based on the points that are most similar. Like the other models, the same train data set was used. Three runs were done using KNN, the first have k=3, then k=4, and lastly k=5. The results of each were appended to a table and an overall accuracy was calculated. From looking at k=3 we can see there was no pattern in the numbers like in Naïve Bayes (*Figure 5).* Overall the accuracy of the model was 92.31%



*Figure 5: KNN with K=3*

**SVM:**

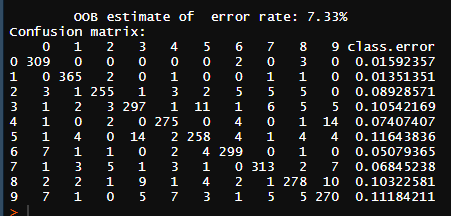
SVM (Support Vector Machines) is a supervised machine learning model that uses classification algorithms for two-group classification problems. The SVM model was run only one. It calculated everything to be a 1. This is only a small sample, and so when an accuracy test was done, it was found to be 12.51% accurate.



*Figure 6: First Run of SVM*

**RANDOM FOREST:**

Random Forest is a type of decision tree. To be specific it look at a cluster of decision trees to make them more precise. The Random Forest Classifier was run once. 500 trees were used to run the modeling. The confusion matrix that it was very accurate in predicting the correct results, with only an error rate of 7.33% (*Figure 7)*



*Figure 7: Random Forest Classifier*

**ALGORITHM PERFORMANCE COMPARISON:**

Overall, the accuracy of the decision tree was higher than the Naïve Bayes Model. The average of the decision trees was 78.32%, where as the Naïve Bayes Model was most likely less than 25% This could have been for a number of reasons, including the training set not being properly used and it being too large to run on the machine. In the future it would be best to further fine tune the models and subset the training set in order for it to run accurately. The SVM model was similar to Naïve Bayes where most of the numbers were predicted to be 1. It was only 12.51% accurate. Random Forest used multiple decision trees to tune the model and was 92.67% accurate, much better than the singular decision tree. The last model was KNN which was 92.31% accurate. Both KNN and Random Forest were the most accurate of the five models.

When comparing the performance between the models, the decision trees took 5.16 seconds to run and the Naïve Bayes model took 14.93 seconds. The decision tree run time was only calculated for Tree 3 as this was the most accurate model created. Since only one Naïve Bayes model was run, I wanted to compare it to only one decision tree model. The random forest ran 60.79 seconds, SVM was 27.82, and KNN ran for 40.34 seconds. The best two models were KNN and Random Forest, and they were the longest running models as well. The Random Forest however ran 20 seconds longer than KNN yet was only 0.30% more accurate than the other.

*Figure 8: Elapsed Time between Models*