Zane Alderfer

Prof. Bolton

IST 707

12/1/23

Homework 6 & 7

Introduction

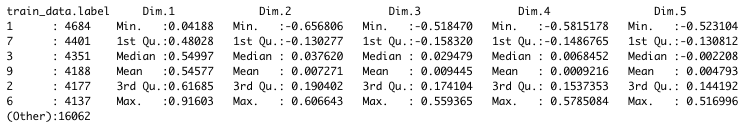
Kaggle possesses a wide variety of interesting datasets for exploration and further analysis. Among these datasets is a dataset solely dedicated to a digit recognizer competition. The basic idea of this competition is to be able to recognize digits 0-9 in a handwritten image. Machine Learning tools and AI can have problems identifying numbers within a picture because these tools don’t see these pictures like human beings do. They view these pictures in the form of pixels and have to look at these pictures one pixel at a time.

This paper will use models that develop guesses for the pictures themselves, the methods used here may vary in results in terms of predicting the correct number. Some of the types of methods used here will be building a decision tree model, a naive bayes analysis, as well as building kNN, SVM and Random Forest Models. The idea here will be to compare the outcomes of all these models to decide which model is the best at predicting the correct number. While some of these methods may be better at predicting the number than other models, that shouldn’t take away from the accuracy of the other models as they could be more effective in other areas while the models used well in this instance may perform poorly elsewhere.

The goal would be to pair some of the methods used such as pairing decision tree model and naive bayes analysis. That would indicate that kNN, SVM, and Random Forest will also be viewed as a group analysis and those results will be compared with the former. Machine Learning tools being able to identify pictures as well as humans would be hugely beneficial in today’s world as this can lead to better image optimization, save time and just give a better overall user experience.

About the Data

The dataset is from Kaggle about digit recognizer labeling from handwritten numbers. The numbers are 0-9 but the dataset is too large to reasonably make models and analysis on so a training and testing dataset were created to amend this issue. The total number of columns for the new dataset is 1400 rows and 1784 columns separated into 5 dims.

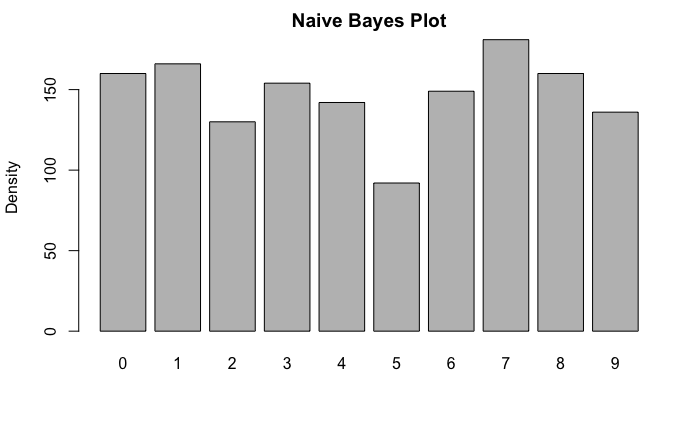


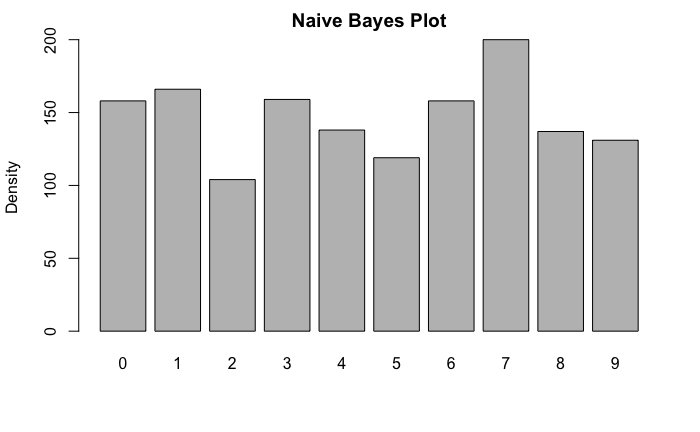
**Preprocessing the Data**

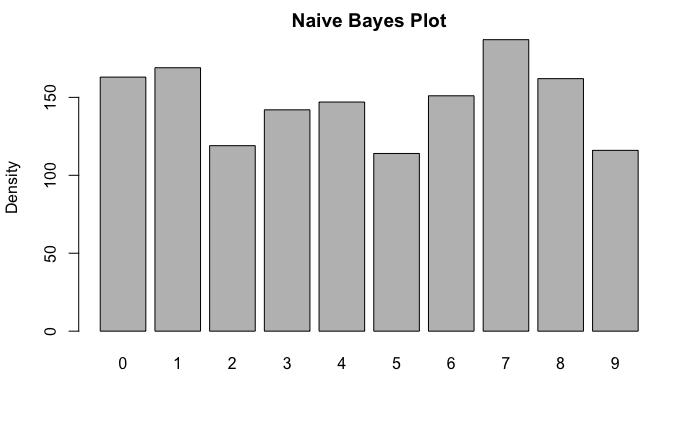
An arbitrary percentage of 28% of the training data is taken and placed into a new dataframe with the digitsplit used as row names. The data label numbers in this training set will be used to develop accuracy results in the test set. The test dataset is dimmed in the same way as the training set for better cross validation. 8 splits were the chosen with variable name “kfolds” to look at the consistency across dataset splits.

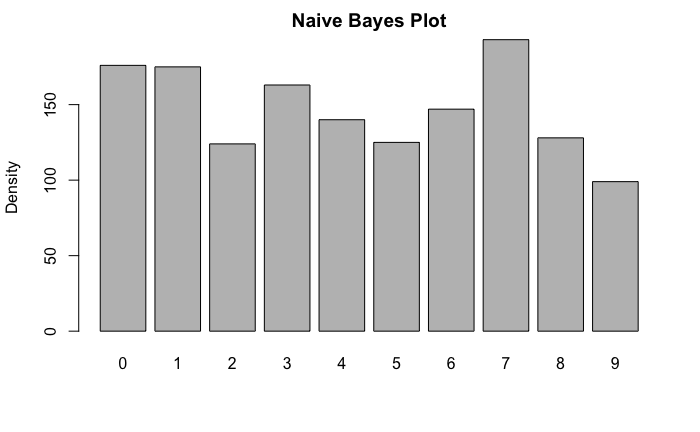
**Naive Bayes**

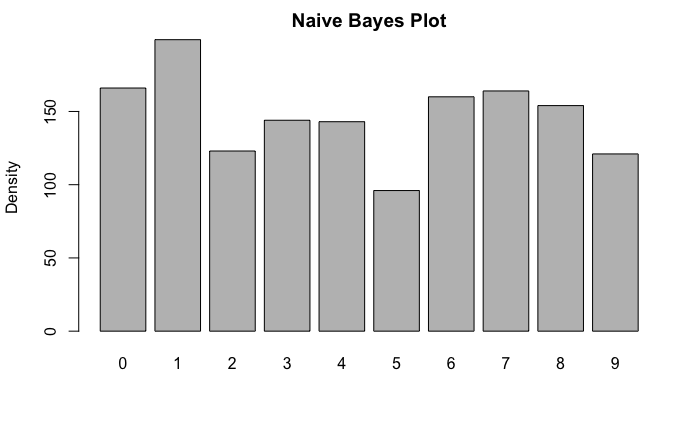
Two empty lists called “AllResults” and “AllLabels” were created to collect the numbers that are correct and the numbers that were predicted by the Naive Bayes model with the corresponding plots showing the results. The training dataset is used first. After using the predict function within the NaiveBayes library, a confusionMatrix function is used to develop the results against the testing data. After choosing 8 folds, the results of all 8 sample datasets are shown below.

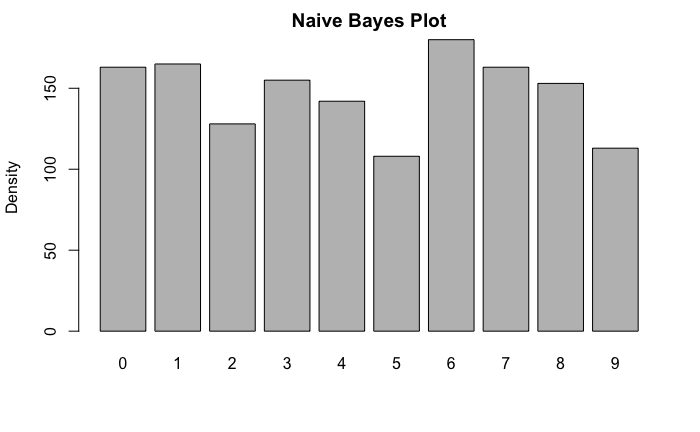


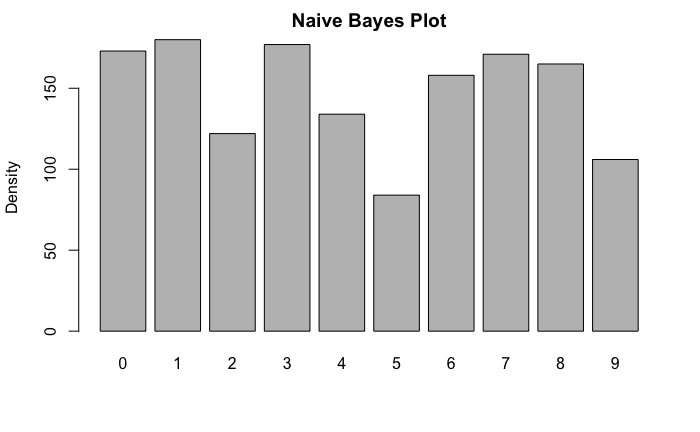






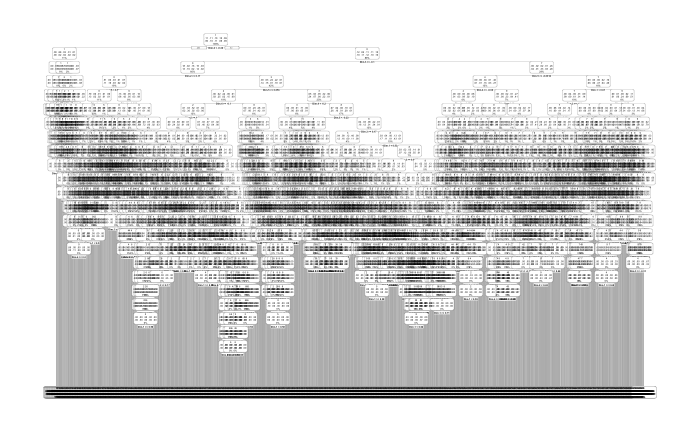




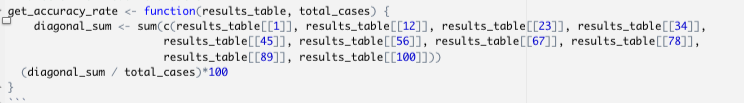


**Decision Tree Model**

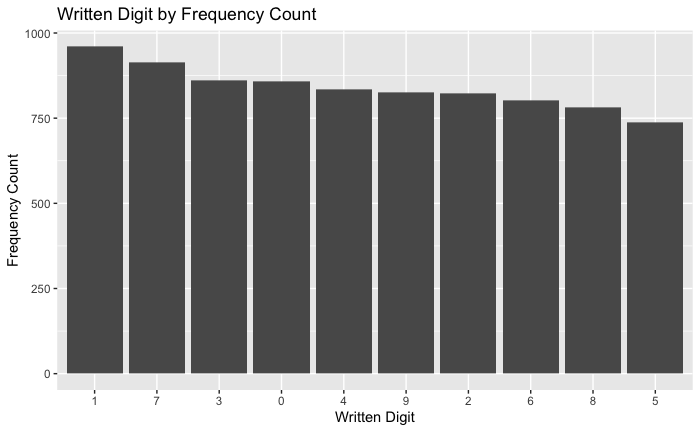
A decision tree model was attempted but due to the volume of the dataset, a lackluster tree was created with too many nodes to comprehend the data as seen below.



Two new datasets were created to work on the kNN, SVM and Forest Tree Models. The label row is factorized and the same dimming process is used again and a function to develop the accuracy rate of each model is created as seen below.

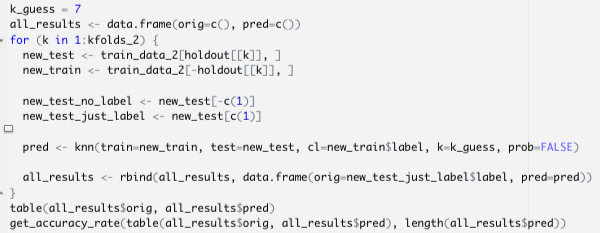


The dataset is then binarized and this allows the dataset to be easily processed due to the simplification of the numbers involved but also helps with the accuracy and efficiency of the machine learning models. Using the SQL library within R Studio, and using this dataset and the count of each number, a ggplot is made for a better visualization of the frequency of each number.

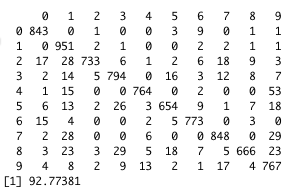


**kNN models**

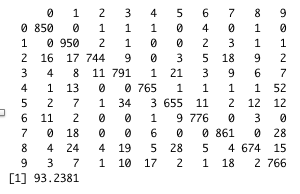
Jumping into the kNN model, four separate models were run with varying folds: 7,3,5, and 8. Using the different amounts of folds, a more accurate depiction of how accurate the kNN model analysis. An example of one of the models can be seen below along with all the results of the models.



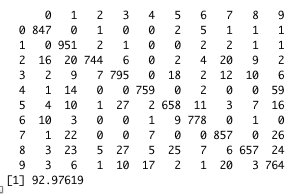
7 folds kNN results



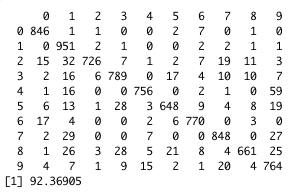
3 folds kNN results



5 folds kNN results

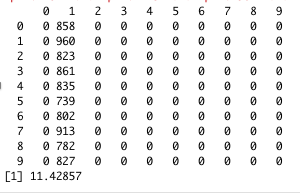


8 folds kNN results

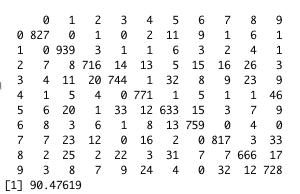
****

**SVM models**

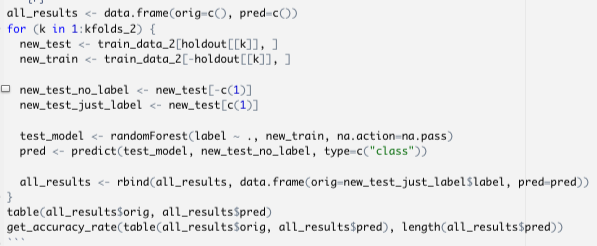
A regular baseline SVM model is created prior to binarized data. The results show why binarized data is important when creating machine learning models that revolve around number imaging identification.

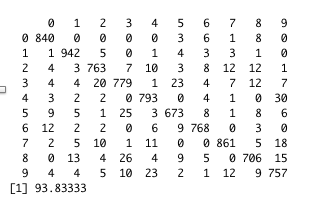


The binarized version of the dataset is created with the results as follows.



The last model to be shown is the random forest model. Using the same outline and foundation of the other models, the results are created below.

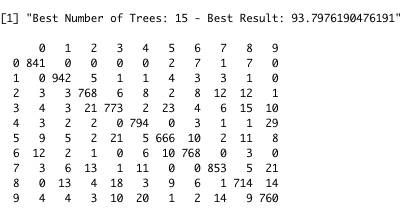




**Random Forest model**

Running the RandomForest model multiple times with a for loop function, the most accurate depiction of the model is displayed after running through the model multiple times as shown below.





Results

The tree model showed no relevant information as the dataset was too big to adequately depict the results. The Naive Bayes plot models showed a lot of the same. These models showed a lot of 1 and 7 results. Then the other end of the spectrum would be a lack of 5’s, 8’s and 2’s being identified. This is reiterated in the frequency count bar graph model created.

The kNN model was run 4 times on 4 separate fold values. The first model was 7 folds and spit out an accuracy result of 92.78%. The second model was 3 folds and gave a result of 93.28%. The third model using 5 folds gave a result of 92.98% and finally the fourth model used 8 folds and delivered an accuracy result of 92.37%.

Looking at the SVM and RandomForest model, a non-binarized model makes it almost impossible for the models to create effective results as when data features aren’t in a binary form, the machine learning models develop a problem identifying the data features. This led to every number in the dataset being identified as a number 1. However, after the SVM dataset became binarized, it delivered pleasing results with a 90.48%. Two versions of the random forest model were created. One using a single run through the model and the second running through the model multiple times with multiple samples to deliver the highest accuracy result. The first run with the singular model delivered an accuracy of 93.83% while the multi-run random forest model gave an output of 93.8% accuracy.

Conclusion

Some initial takeaways from the models would be that some of the models aren’t ideal as can be seen in the decision tree model. While there may be a better way of cleaning or pruning the data in such a way that would more effectively show the decision tree model, this model probably wouldn’t necessarily be very effective to begin with as a range of 10 numbers is asking for a cluttered and disorganized decision tree model.

The Naive Bayes models used a different method of displaying results so while they could be as accurate as the latter three models, it’s hard to tell initially. However, the results across the bar graph plots are consistent with the samples used. This would suggest that the results show a level of accuracy that’s respectable otherwise the bar graphs would depict a level of inconsistency across the model results. Due to the relatively similar counts across numbers such as 1 and 7, this could be explained by the similarities between the two numbers. This can also be seen with numbers like 8 and 3 for obvious reasons. It’s also interesting to note the lack of identifying 5 and 2 as this could be explained by the relative complexity of writing the two numbers.

Moving on to the kNN model, the amount of folds seem to show relevance as the lower the number of folds, the more accurate the results seem to be. This is probably due to the more folds used for training, the more the training models will vary which could impact the testing data. If the training models have more data, the law of large numbers can begin to take over and eventually reach its mean (higher accuracy). When compared with the binarized SVM model, this had a slightly lower accuracy with only 90.48% while the best kNN model was slightly over 93%. The last model used was the Random Forest model. Interestingly enough, the model that was only run once had the best accuracy of all the models at 93.83%. The Random Forest model that ran multiple times and identified the best number of trees had its highest accuracy at 93.8% which is slightly lower than the first model run. That number of trees being 15.

All of these models have merit and could all be used effectively in specific circumstances such as the decision tree model with the disputed papers. But in this case, the best model to run seems to have been the Random Forest Model.