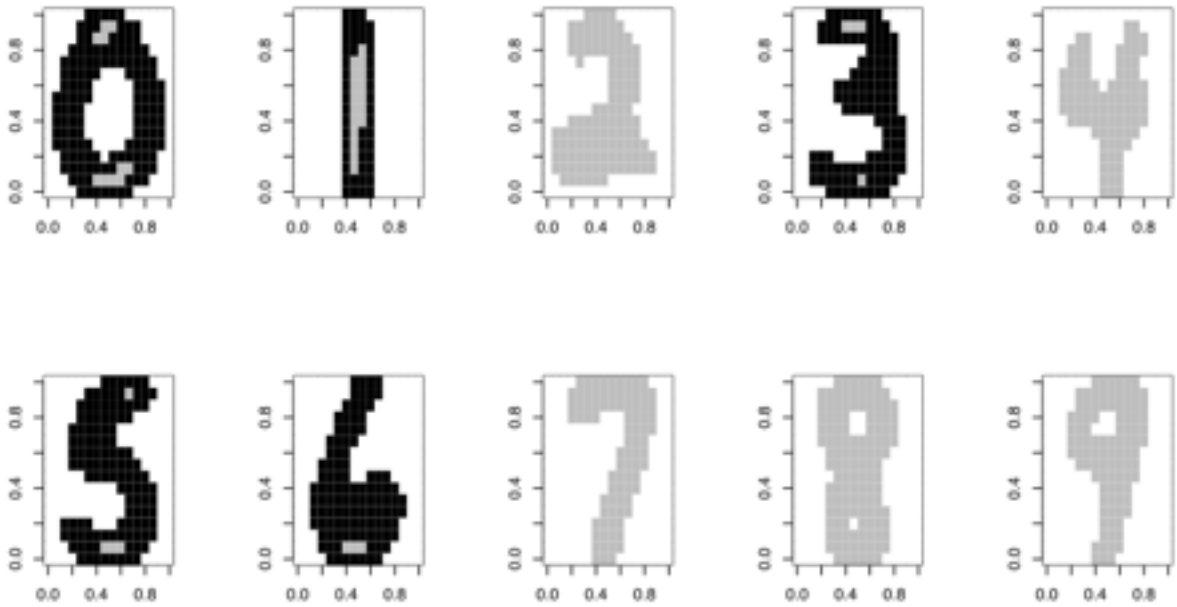
STA 141A Final Project

12/4/17

Q1 – See code

Q2 – See code

Q3 – This is what each digit looks like on average.

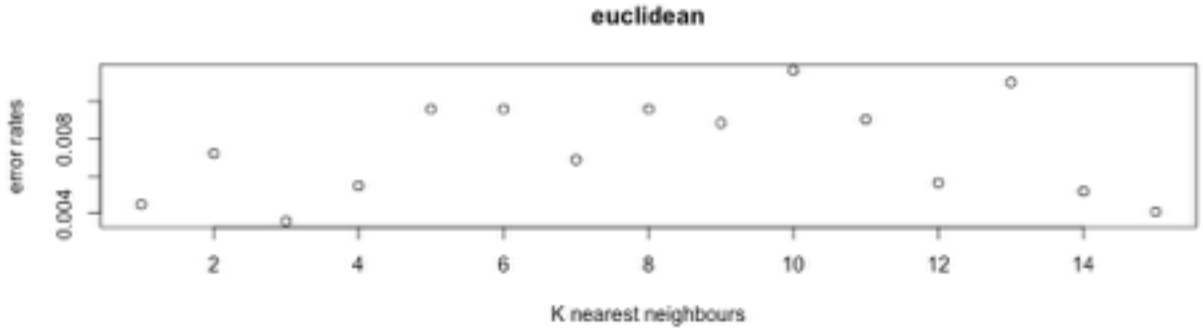


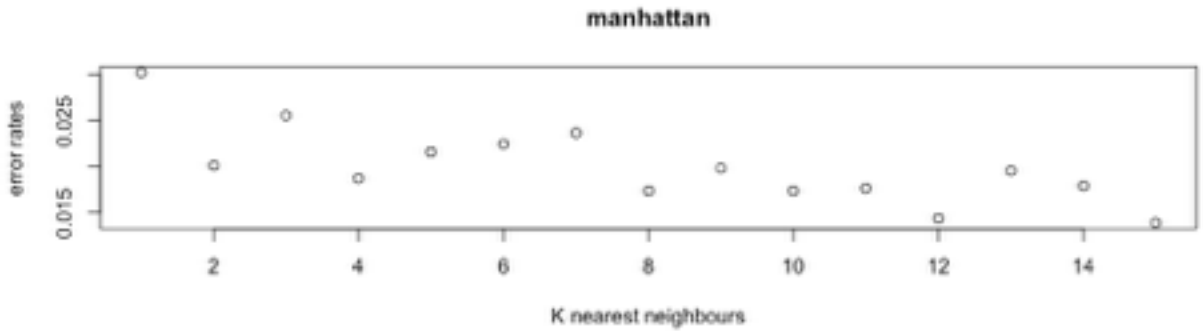
The pixels that are most useful for classification are the dark pixels that are used to define the perimeter of the digit image. While the pixels inside the perimeter do bear some importance they are not nearly as important as the pixels forming the silhouette.

Q4 – Refer to code

Q5 – able to successfully write a function that used 10 fold cross validation estimate the error rates of k-nearest neighbors. At first, our function was inefficient and took too much time to run. One strategy that we used to make the code more efficient was to break up the components of the function into separate functions which helped us avoid looping through the entire dataset. Additionally, we also subsetted the data into 10 folds by using the which function which helped us quickly segment the data and create the model based on only the indices that were required which saved time.

Q6 – The following plot is a simple plot of the estimated error rates of k nearest neighbors using Euclidean and Manhattan distances.



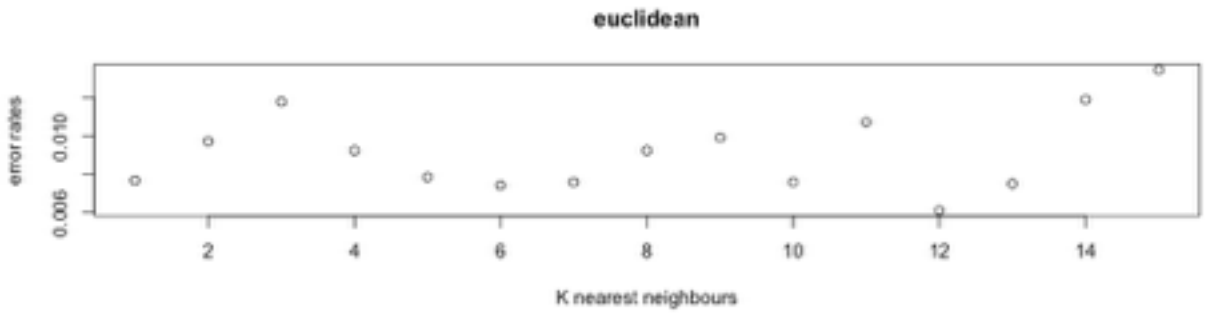


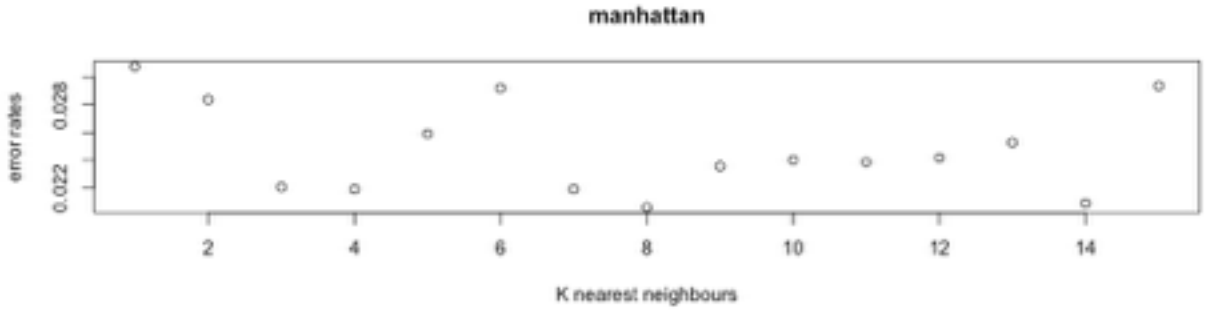
In general, Euclidean distance has lower standard error rates when using our model. However, it appears that lower and high values for k tend to perform better for Euclidean distance. In our test our best k value was k = 1 for Euclidean distance. It is possible that our function would reproduce different error rates after running the function multiple times, however our error rate for k = 1 is so low that we believe it is the most accurate. I do believe it would be useful to consider additional values of k in this dataset because it is a larger sample size.

Q7 - Our best three k values were k = 1, k = 3, and k = 15 for Euclidean distance. We were able to estimate a confusion matrix. Our confusion matrix yielded 97.01% accuracy for k = 1. For k = 3 we had 96.65% accuracy. Lastly, for k = 15, we were 95.2% accurate. Thus, we would not change the combinations of k and distance metrics and we still believe k = 1 with Euclidean distance is the best combination.

Q8 - Our model falsely predicted 8 the most often. When examining the data, 8 was confused with 3 16 times. Additionally, our model frequently confused 2 with 7. The 2 and 7 classifiers are actually very similar. They have similar grayscale images and so this makes sense. The 8 and 3 have similar shapes (the 3 is sort of like on side of an 8 classifier) however have completely different grayscale values on average which raises concerns on how the model performs.

Q9 – The following plot is of the test set error rates for all k values 1 through 15.





The results follow similar trends to the 10-fold CV error rates however the test results have generally higher standard error rates. There are some noticeable differences. For example, the 10-fold CV error rate for k = 15 was very low compared to the other values and in our test data k = 15 actually has the highest error rates. Additionally, for Manhattan distance, k = 8 has the lowest error rate in our test data. In the 10-fold CV error rate, k = 8 was relatively in the middle of the road. In our test data, there is also large differences between the two metrics. Or example, Euclidean distance has k = 3 at a higher error rate compared with it's immediate neighbors whereas Manhattan distance has k = 3 at a very low error rate compared to it's immediate neighbors.

Appendix

Q1 - No citations

Q2 - No citations

Q3 - No citations

Q4 - No citations

Q5 - Help with cross validation from: http://genomicsclass.github.io/book/pages/crossvalidation.html

Q6 - No citations

Q7 - Help with confusion matrices from: https://stackoverflow.com/questions/26631814/create-a-confusion-matrix-from-a-dataframe

Q8 - No citations

Q9 - No citations