ISYE 6501 – Homework 1

Question 2.1
I work for a political campaign. A Classification Model would be appropriate for many different situations, but one good example is in determining whether or not someone would potentially contribute to the campaign. Potential predictor could include income level, previous political giving history, age, occupation, or education levels.

Question 2.2.1
Please review the included 2.2.1.R file along with this write-up, which includes the code I used and comments on what each section is doing.

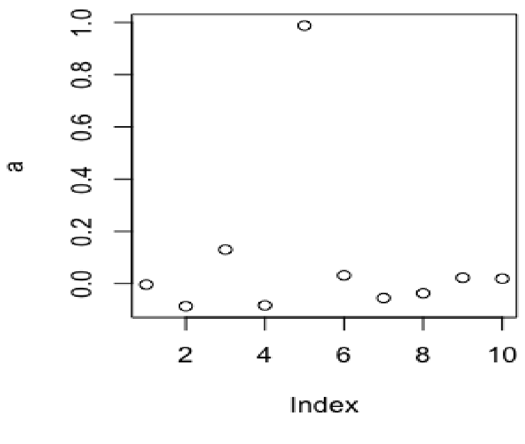
Using the attached code, I modified the C value from a range of .00001 to 1,000,000. Changes to this value changes the models tradeoff between minimizing classification error and maximizing margin. I chose a C value of 100,000 because it resulted in the minimum classification error with the highest margin. Margin for this model is 1.986 and accuracy is 86.39%.

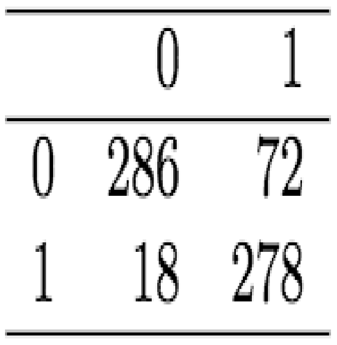
The equation for the classifier at this C value (rounded to 3 decimals to save space) is:

$$(-0.004 * X1) + (-0.088 * X2) + (0.130 * X3) + (-0.083 * X4) + (0.988 * X5) + (0.031 * X6) + (-0.056 * X7) + (-0.037 * X8) + (0.022 * X9) + (0.019 * X10) + .081 = 0$$

Because changes in C values over a large range (between .001 and 100,000) have minimal impact on the model's output, we can guess that the data is likely well separated and we don't have to make much tradeoff between a large margin and avoiding mistakes.

I also plotted each a value to see which features in the model had the highest predictive value. Surprisingly, most of the weight in the model is on factor 5 (x5), as shown in the following plot. Additionally, by plotting a confusion matrix of factor 5 vs. the correct outcome, we can see that factor 5 is predictive in about 86% of cases (286+278/654=.862), very close to what our SVM predicts. This may mean that we could ignore other factors when building a production model.





	0	1
0	286	72
1	18	278

Question 2.2.2
I tried a few different kernels, but found the highest accuracy with the radial basis function kernel ("rbfdot"). As shown in the attached 2.2.2.R file, prediction accuracy is over 99.5% with this kernel with a C value of 10000. This is almost certainly overfitting to a large degree. We are using a non-linear kernel which increases the model's ability to separate linearly non-separable features. Additionally, by using a high C value, our model is

favoring minimizing error over margin increasing the likelihood of overfitting. Most importantly, by testing accuracy on the same dataset we used for training we have no way of validating that the model is not overfit to the training data. It would likely perform worse on a new data set.

Question 2.2.3
Please review the included 2.2.3.R file along with this write-up, which includes the code I used and comments on what each section is doing.

After loading data, I used the train.kknn function to find the best k value between 1 and 100. Train.kknn iterates over k values from 1 to a chosen max, using leave one out cross validation. This avoids the issue mentioned in the homework of including the current value of i when training the model, because it tests each i against all the other data points. The resulting best k value was 22. I then then divided the data into train and test data sets by randomly sampling 80% of the data into the training set and using the remaining 20% as test set.

I created the model with a k of 22, fit the model, and converted the continuous results into binary results by rounding. I saved the resulting predictions and generated a confusion matrix, showing the predicted vs. actual results. Finally, I computed the model's accuracy by summing the correct predictions and dividing by the total number of rows in the dataset. Accuracy with a k a 22 was 87.78%, slightly better than with the SVM from 2.2.1.