Lee Sutton
301145106

CMPT 726 Assignment 2

Problem 1 Linear Models for Classification

To classify these three regions, we can use 3 linear functions such that:

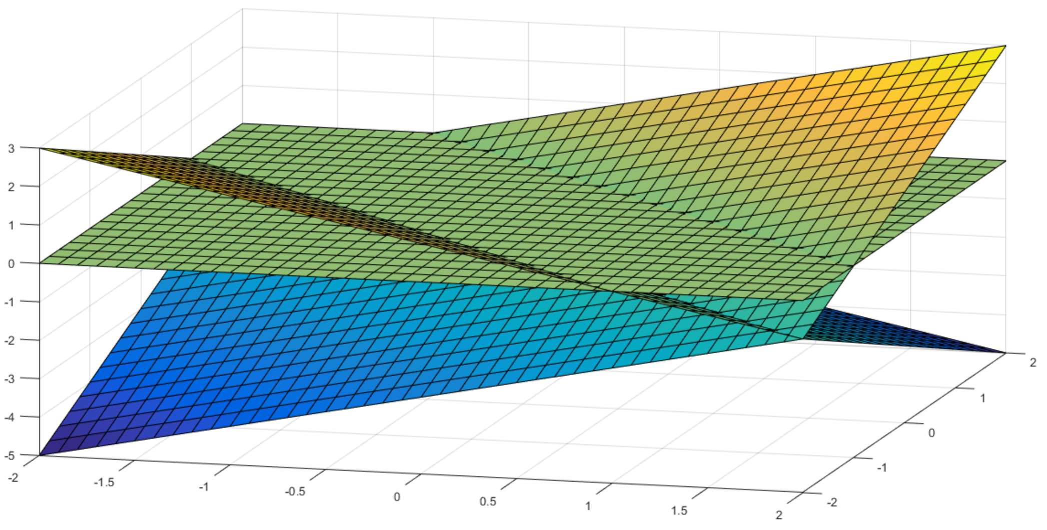
$$y_1(x) > y_2(x) \; and \; y_1(x) > y_3(x) \; for \; x \; belonging \; to \; class \; 1$$

The same applies to class 2 and class 3. The following equations achieve this based on the given decision boundaries. These surfaces were also plotted in Matlab and can be seen below in the figure.

$$y_1(x) = -x_1 - x_2 - 1$$

$$y_2(x) = 0$$

$$y_3(x) = x_1 + x_2 - 1$$



Problem 2 Kernels

1. The kernel functions and coefficients were formed using a dual formation on the least squared error solution. As a result, the coefficients found using the kernel regression will give the same function as the coefficients found using linear regression. However, if regularized regression is used to find the coefficients, the kernel regression will give a different final function. The regularized regression penalizes large coefficients w, but with kernel regression the weights created by the kernel multiplication will not be included in the error function. For example, in the polynomial kernel corresponding to $\boldsymbol{x} \rightarrow (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2)$, the $\sqrt{2}$ will not be

included in the error function. As a result, the regularized regression will give different results for the kernel regression with the polynomial kernel.

2. Given two kernel functions:

$$K_A(x,z) = \varphi^A(x)^T\varphi^A(z)$$

$$K_B(x,z) = \varphi^B(x)^T\varphi^B(z)$$

A third kernel function can be constructed by using:

$$K_C(x,z) = \alpha K_A(x,z) + \beta K_B(x,z)$$

It can be shown that K_C is a valid kernel since it corresponds only to dot products

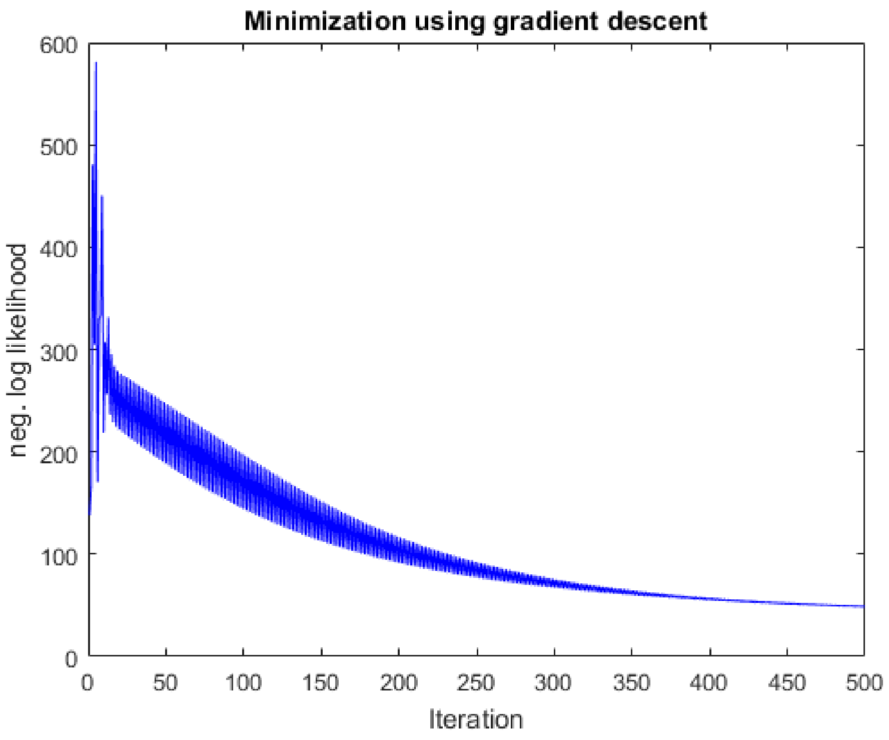
$$K_C(x,z) = \alpha\varphi^A(x)^T\varphi^A(z) + \beta\,\varphi^B(x)^T\varphi^B(z)$$

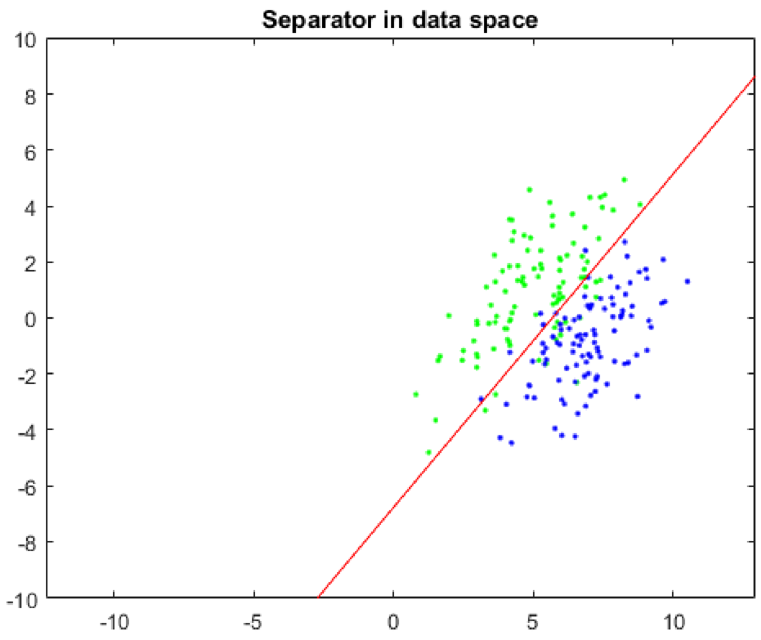
Since both α and β are positive, it can also be shown that the gram matrix is positive semi-definite. By definition, this makes $K_C(x,z)$ a valid kernel.

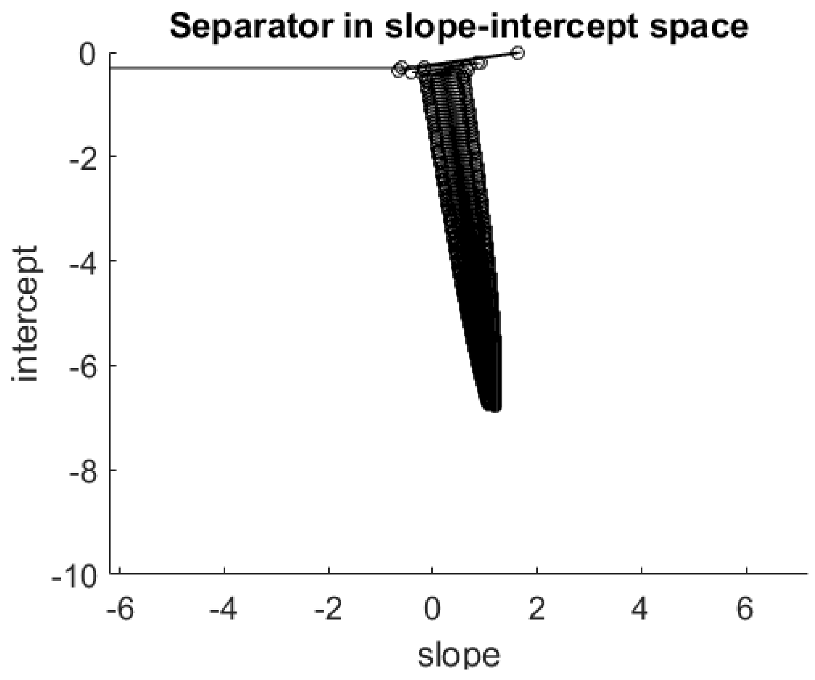
Problem 3 Logistic Regression

1. The plots produced by logistic_regression.m are shown below. When running the code, the line separating the data would oscillate back and forth around the middle of the data. The gradient descent method evaluates the gradient and the current value and then steps a finite amount in that direction. Since this step size is finite, the line oscillates around the minimum value, constantly stepping past it then turning around and stepping past the minimum again. This causes the line separating the data to oscillate. The final results are shown below.

Lee Sutton
301145106

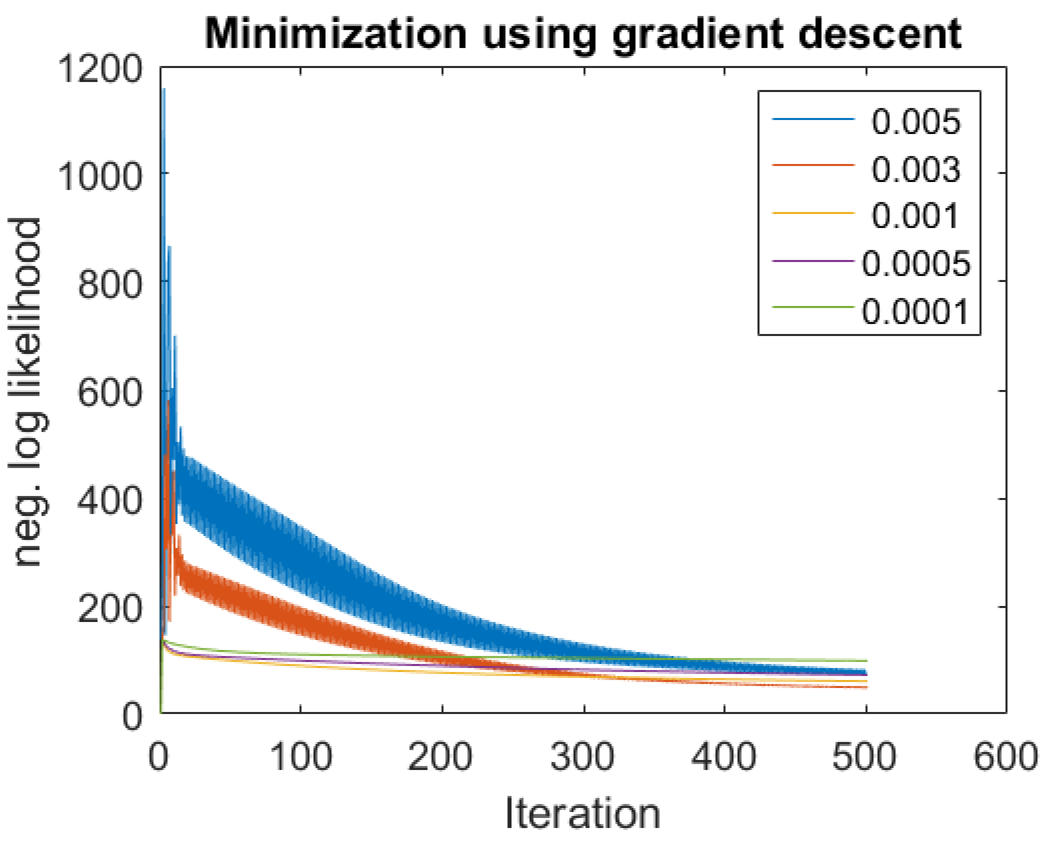






2. For problem 2, the logistic_regression.m script was modified so that the step size varied from 0.005 to 0.0001. The results are shown below. It appears as though a moderate step size produces the best results. A step size that is too large descends quickly but can oscillate around the minimum and does not produce the smallest error. If the step size is too small, the system moves slowly. The gradient pushes the system towards the minimum however, due to the small step size, it takes more iterations to reach the minimum. A smaller step size might produce more accurate results and will show less oscillation around the minimum however, it takes more iterations to get there. A moderate step size produces a balance between speed and accuracy. The system will descend at a moderate pace towards the minimum and show small oscillations around the minimum.

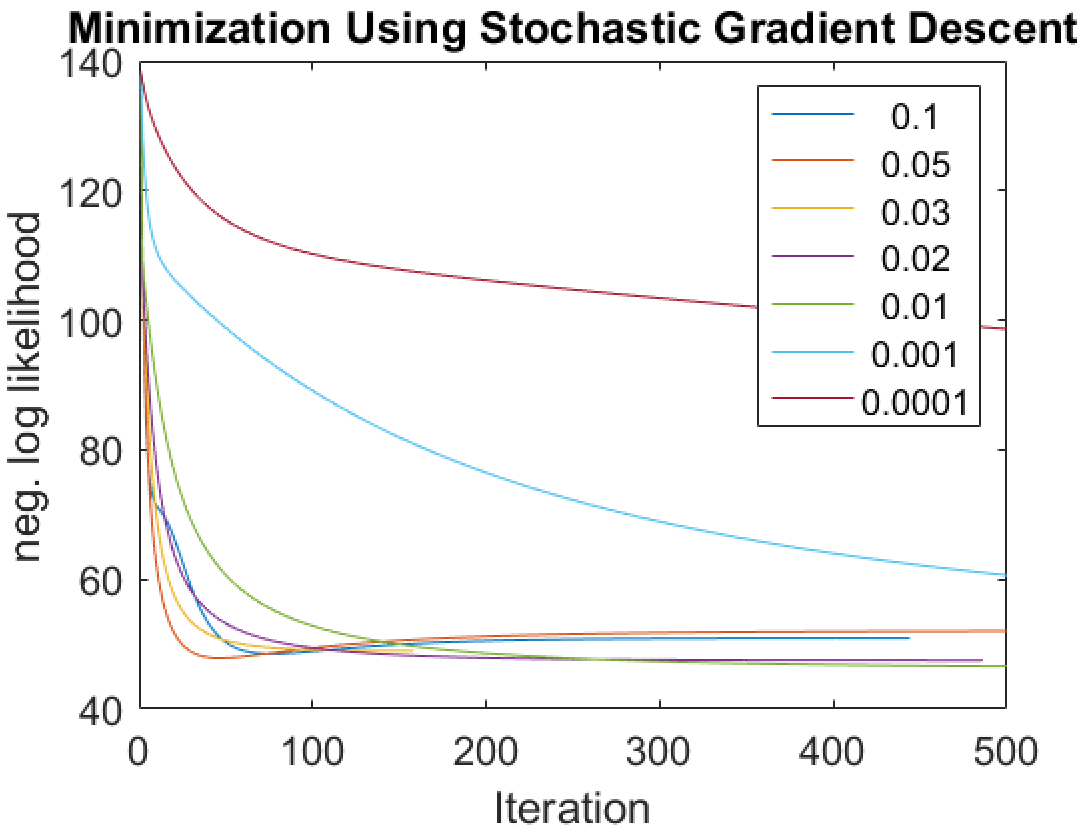
Lee Sutton
301145106



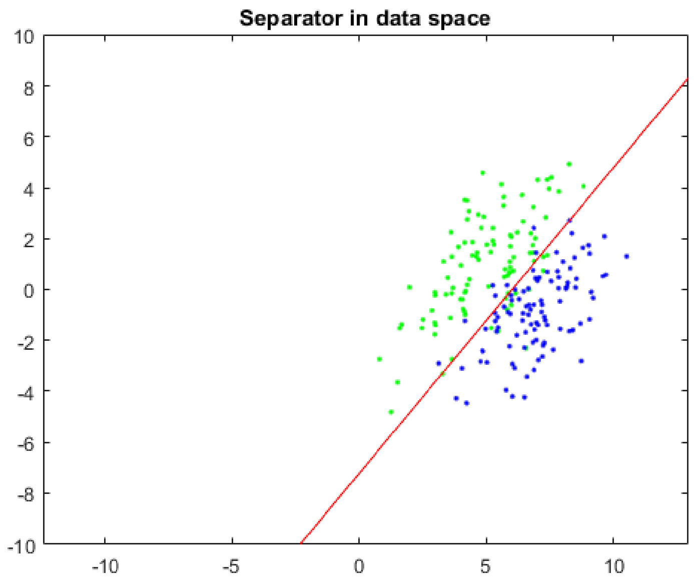
3. For problem 3, the logistic regression script was modified to use stochastic gradient descent. This method uses the same concepts as gradient descent however, instead of taking the gradient of the entire function, stochastic gradient descent calculates the gradient at a single training example and updates the function. This process is continued either 500 times or until the tolerance is below a certain level. The results are shown below.

   The stochastic gradient descent method worked faster than the gradient descent method. This can be seen in the graphs. For all step sizes, the error function for the stochastic gradient descent reached a lower error level then the error level on the gradient descent graph in less iterations.

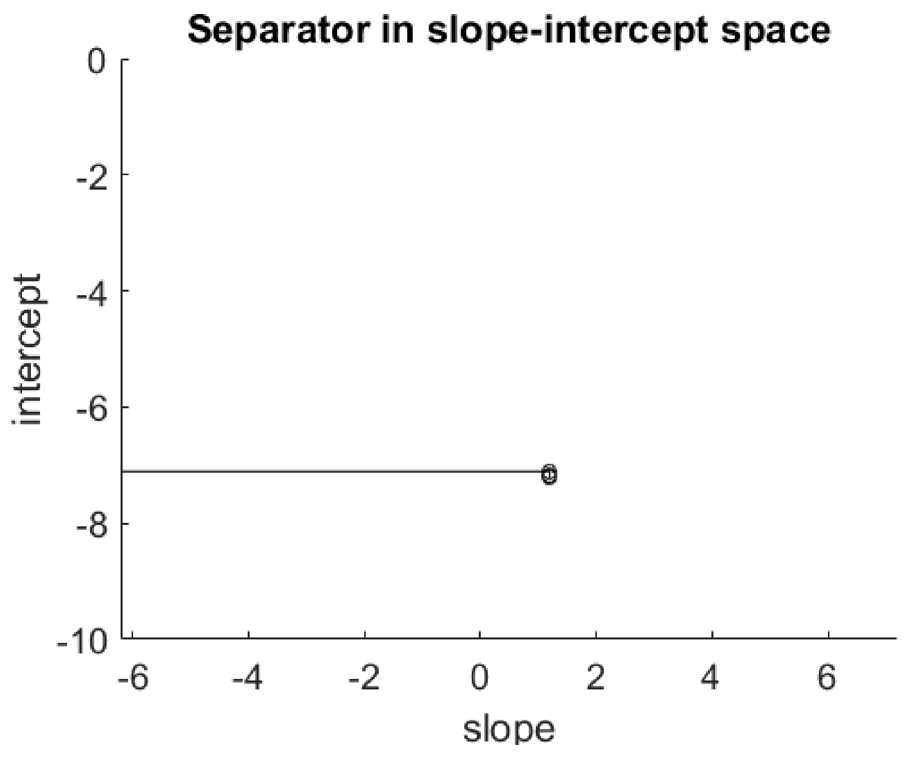
Lee Sutton
301145106

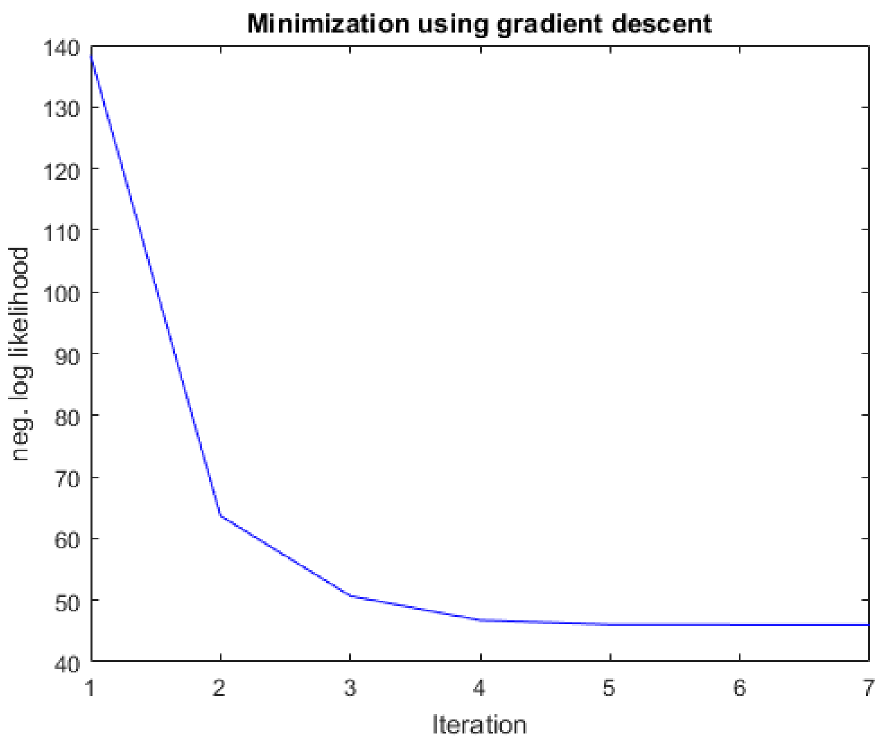


4. For problem 4 the logistic regression script was modified to use iterative reweighted least squares. The algorithm finished very quickly, requiring only 7 iterations. The results are shown below.



Lee Sutton
301145106





Lee Sutton
301145106

Problem 4 Kernelized Perceptron

For the spam email problem various kernel functions were tested using cross validation to determine which function produced the best filter. The following kernel functions were implemented and tested.

1. Gaussian Kernel

$$K(x,y) = \exp\left(-\frac{||x-y||^2}{2s^2}\right)$$

2. Polynomial degree 1

$$K(x,y) = (1 + x^T y)^1$$

3. Polynomial Degree 2

$$K(x,y) = (1 + x^T y)^2$$

4. Polynomial degree 5

$$K(x,y) = (1 + x^T y)^5$$

5. Sigmoid Kernel

$$K(x,y) = \tanh(ax^T y + C)$$

6. Exponential Kernel

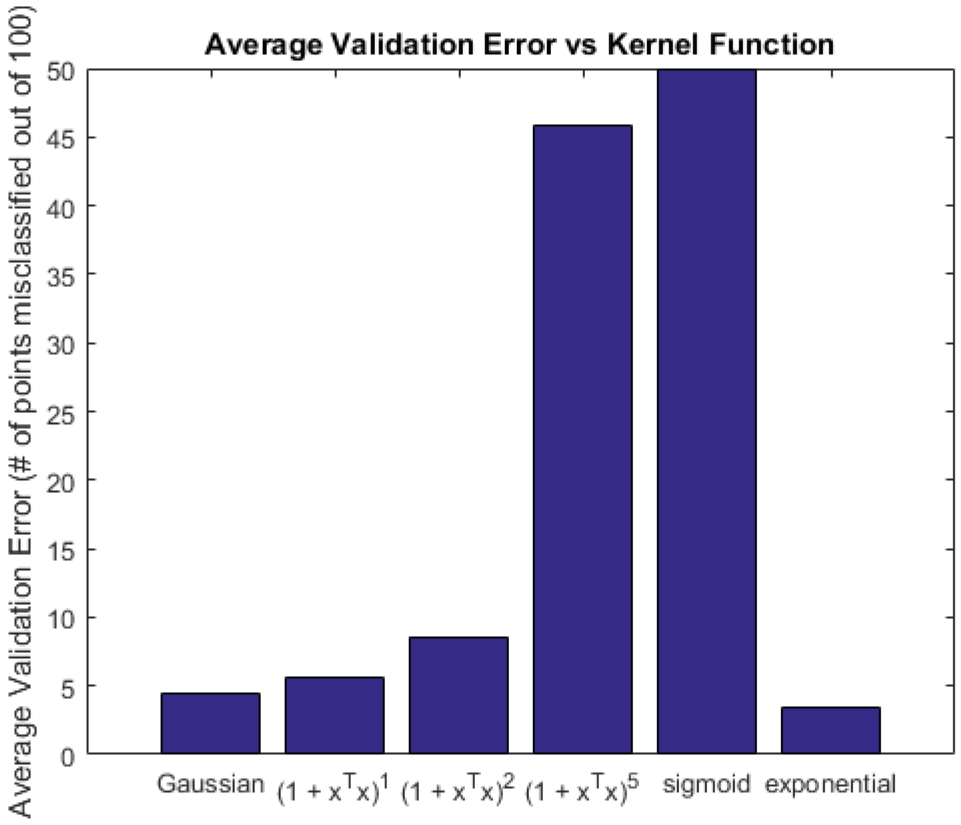
$$K(x,y) = \exp\left(-\frac{||x-y||}{2s^2}\right)$$

The training data was then split into a training set and a validation set. The validation set contained 100 randomly selected data points such that 60 came from ham, and 40 from spam. Each kernel was tested 5 times with the validation error recorded each time. The average over these 5 trials was then taken to find the best kernel function. Although there were many other factors that could have increased the effectiveness of the algorithm, including step sizes, validation selection, iterations. Most of these were held fixed as considering changes in this data was too much to consider to effectively build an email filter in the given time frame. The final validation results are shown below. Although, the Gaussian, the 1st degree polynomial, and the exponential function gave similar results, the exponential function gave the lowest validation error. The exponential function was then used on the test data. With the following parameters.

Lee Sutton
301145106

$$K(x, y) = \exp\left(-\frac{\left||x - y|\right|}{2(5)^2}\right)$$

	Gaussian	Polynomail degree 1	Polynomail degree 1	Polynomail degree 1	Sigmoid	Exponential
Trial 1	3	5	10	44	40	0
Trial 2	7	6	10	36	40	8
Trial 3	3	2	5	53	40	5
Trial 4	5	7	5	49	90	3
Trial 5	4	8	13	47	40	1
Average	4.4	5.6	8.6	45.8	50	3.4


Average Validation Error vs Kernel Function
Average Validation Error (# of points misclassified out of 100)
Gaussian
$(1 + x^T x)^1$
$(1 + x^T x)^2$
$(1 + x^T x)^5$
sigmoid
exponential