**Assignment 2**

**Text Classification**

Classifying emails as spam or ham was implemented using two methods. The text files containing both the emails were used, some for training and the rest as test data. Based on the words, their counts and their probability of occurrences, they were classified as spam or ham emails. This was implemented using Java.

**Logistic Regression:**

In logistic Regression for text classification, the training data is used to calculate the weights and these weights are further used while testing a new data for calculating the probabilitites. Overfitting occurs if a model is trained too much it will fit the training data too well but when applied on a new set of data the prediction accuracy is too less. In order to avoid overfitting, we use L2 regularization.

L2 weight regularization penalizes weight values by adding the sum of their squared values to the error term. To summarize, large model weights can lead to overfitting, which leads to poor prediction accuracy. Regularization limits the magnitude of model weights by adding a penalty for weights to the model error function. L2 regularization uses the sum of the squared values of the weights.

**Regularization Weight**

There are several ways to find a good (but not necessarily optimal) regularization weight.  The major advantage of using regularization is that it often leads to a more accurate model. The major disadvantage is that it introduces an additional parameter value that must be determined, the regularization weight.  Increasing lambda results in reducing overfitting but in turn increases the bias.

* Run the code for different values of lambdas.
* For each lambda, calculate the corresponding weights.
* Choose the lambda values for which the weights fit the data properly.

In this code, I have taken different values of lambda such as 0.3, 0.38, 0.4

**Iterations:**

The number of iterations also affects the accuracy of the training data. If the number of iterations are too less, then the data will not even converge. If the number of iterations are too high, then the data will diverge too much. Hence the number of iterations have to be kept moderate in order to get a good accuracy.

In this code I have taken the number of iterations to be 100 which is not too high or too low.

**Naïve Bayes**

Multinomial Naïve Bayes was implemented for text classification. Laplace Smoothing is done while implementing this, so that the zero that can be obtained while calculating the result, does not affect the probability of the words. An extra 1 is added to the numerator and the denominator to avoid ending up in a zero probability. It basically adds one extra count to each word count. This addition will not affect the end result as it is negligible when compared to the actual word count.