**Sinan Yaman**

**0049719**

**Introduction to Machine Learning**

**Homework 1 Report**

First of all, I grabbed the data from the “cvs” files by the code read.csv() and using the parameters “skip=” to assign the first 25 rows to the training matrix and last 14 to the test matrix for each letter data. Then I bound the training matrix of each letter with the rows to form the training set and bound the test matrix of each letter with the rows to form the test set. P(y=c) parameter was the same 1/5 since all the letters had 39 data to the total of 195. To calculate the sample means, first I found the total of every value of each pixel for every row and divided by the total (which is 25) to form the sample mean matrix for each letter. The reason I did all these steps for each letter individually was to see what the result of each operation is more clearly. Again, sample mean matrixes were row bounded to form the pcd matrix. To calculate the score matrix, I first defined a safelog function which returns 0 if the argument is zero and log of the argument otherwise. Used the formula Sum from x=1 to X, xd\*log(pcd) + (1-xd)\*log(1-pcd) + p(y=c) and calculated the 125x5 score matrix that shows what each row of each letter scored for the letters A through E. This is done with three for loops and the score matrix is formed. Next, I found the max of each row and assigned the index of the max to the estimation matrix which became a 125x1 matrix. Truth matrix was already given in the labels.csv file, each letter has 25 entry. The confusion matrix was formed using the table() function. Same process is done with test data also, but instead of 25 rows we had 14, therefore matrices were 70x1. Algorithm worked quite well, apart from the letter “E” which was confused a lot as we see both in the confusion matrix for the training data as well as the test data.