Blog

For this assignment, I took a lot of time working through the code from lecture to figure out how everything works. When I initially ran the PAA from class, the output seemed to be very random. Using the matlab function normalize() seemed to make the visual representaion make more sense.

The next challenge I faced, was trying to get the confusion matrix to run properly. The parameters for the confusion matrix were the training data and the test data and another parameter I did not understand at first. That parameter ended up being a column of where each class was located. I solved this by hard coding a 1x600 line of class numbers 1-6 based on where they should be for the data set. An additonal parameter I added was the distance function. This distance function could be input into the function with the @ operator. To get the training data parameter, I took the mean of each column within each class to provide a 6x60 matrix of training data. Each row represented the averaged values of each class from the data set.

At this point the confusion matrix would output, but the data did not make sense. The manhattan and euclidean distance formulas appeared to be providing the same output. I realized that what was being input was a single data point from both the training data and the test data. The expected input should have been the whole row. The purpose of this distance calculation was to see how close the row compared to the training data. This is why the mean of the data was used to create the training data, so that the result would be more consistent.

The final challenge was figuring out SAX. I honestly don't think I did it in a way that was intended as I was only able to create the graph and did not create an actual data set. The plot was created by adjusting the paa plot function. I defined that for every value that rounded to the nearest .5 would get a new label. The labels were A - M that were used for -3 - 3. The output looks as it should, but I am still unsure how to get the actual SAX data set.