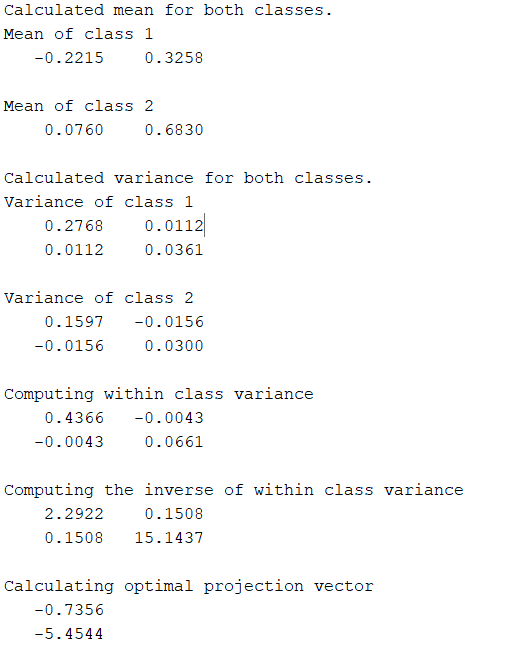
**PROGRAMMING ASSIGNMENT – 1**

**1)Implementing fisher LDA**

**Folder – LDA Implementation File LDA.m**

Have implemented LDA by accessing the given riply data and dividing the dataset according to labels. Once respective Class1 and Class2 are obtained have obtained mean of both class matrix and followed the other claculation like varience, varience within the class then inverse of that and finally optimal projection.

**The obtained results in console are as below.**



**2)Implementing** **perceptron**

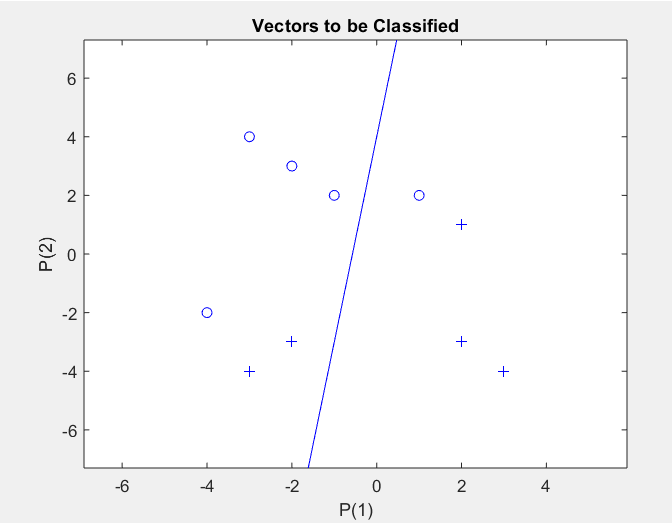
**Folder – LDA Implementation File LDA.m**

Have implemented perceptron by intializing weight and biases, then multiplying weights with the data to check resultant Yi is greater than or equal to zero and then we compare the output with desired output. If the output is different from desired one we claculate error and update the weights accordingly. So that for each iteration the weights get updated and the hyper plane changes to divide classes properly.

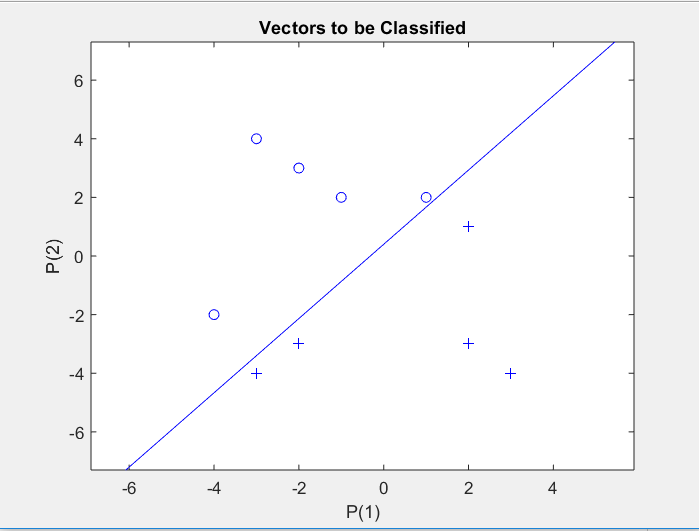
f(x) => 1 if W.X+b >= 0

f(x) => 0 otherwise

**The updated graph once iterating all the dataset**



**The final graph once it iterated certain number of epochs .**



**3)** **Generating synthetic data to run on both algos**

**FUNCTIONS USED:**

* **RandomDataGen:** This method generates random synthetic data and respective label vector which can be run on both our algorithms.
* **FisherLDAFunc:** This function implements the Fisher LDA method. It takes the data generated from the RandomDataGen and gives optimal projection vector and threshold value.

**Inputs:**Data,Labels

**Outputs:** v,Threshold value

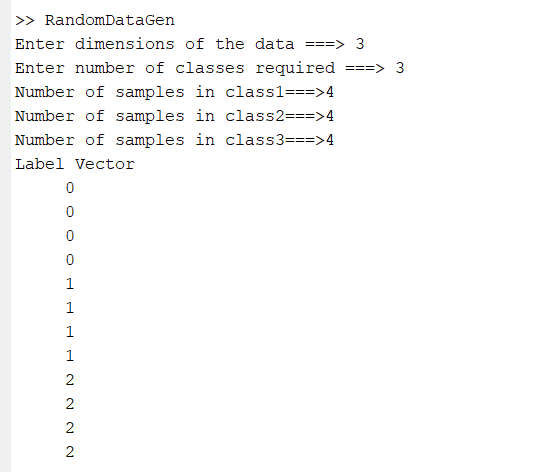
* **PercepFunc:** This method again takes the user generated data as input and finds hyperplane to divide them.

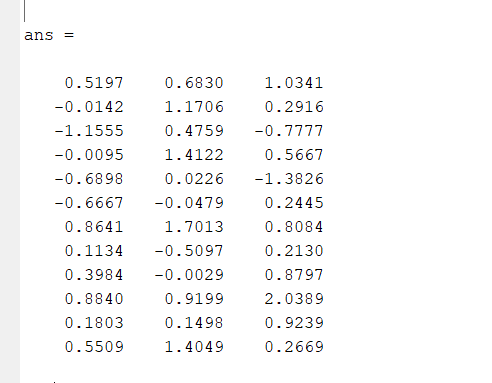
**Inputs:** Data,Labels

**Outputs:** UpdatedWeight, UpdatedBias

* **RunningBothFuncOnUserGenData.m** This file is used to run all of the above files and print generated results.

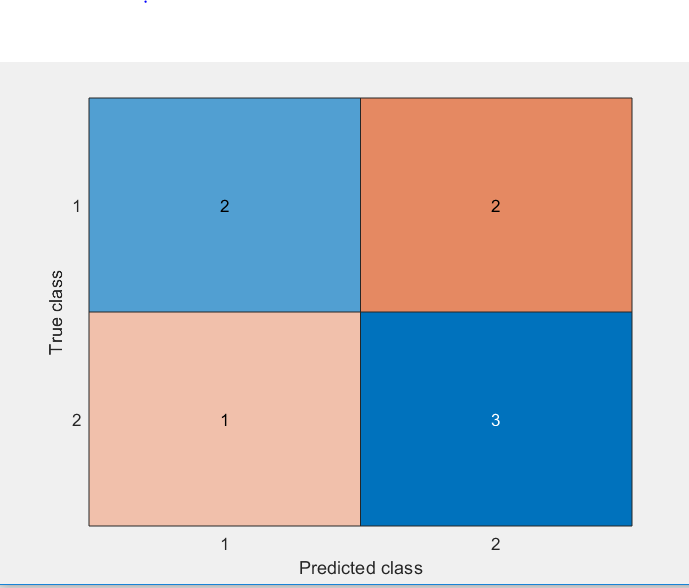
**The Output for randomDataGen file we specifiy the dim and classes then number of instances and it gives us randomly generated matrix of dataset with labels.**





**Evaluation of algorithms using confusion matrix**

Implemented confusionmat to plot the confusion matrix and then confusionmatStats fuction gives the sensitivity, specificity and accuracy.



**4)Validation with Synthetic Data**

Have implemented both LDA and Perceptron in function (**FisherLDAFunc** and **PercepFunc**) and called them using **RunningBothFuncOnUserGenData** to runSynthetic generated data using **RandomDataGen** and the finally invoked **confusionmatStats** to evaluate the performance of the models.

