Programming Questions – Homework 2 (Saket Vishwasrao)

1) GenerateSyntheticDataL.m generates a linearly separable dataset and GenerateSyntheticDataNL generates a nonlinearly separable dataset.
2) Files used:
   PerceptronUpdate.m – updates the weight using current model and input data and returns the new model
   trainPerceptron.m—trainsPerceptron over the data set and plots the training and testing accuracy in each iteration.
   testPerceptron.m—Predicts output for a dataset given model and calculates it output.

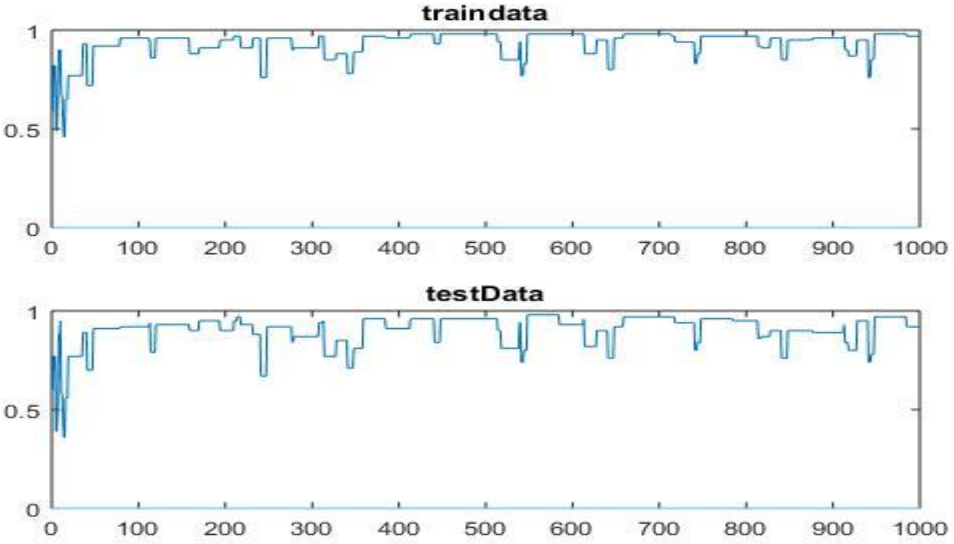


Figure 1: Perceptron training over 10 iterations for Synthetic linearly separable data.

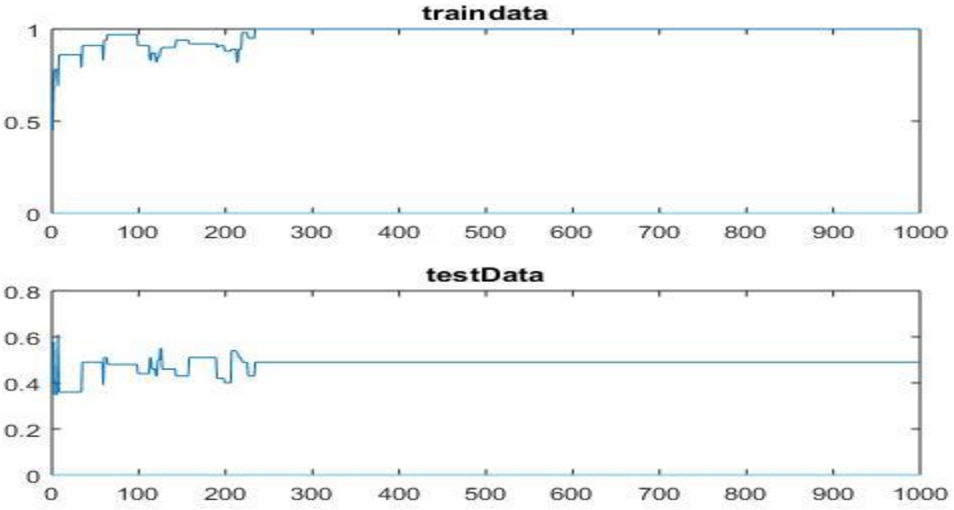


Figure 2 : Perceptron training over non linearly separable data.

Discussion:

From figure 1 and figure 2, it is apparent that the perceptron gives a very high accuracy (~95%) for the linear test data and a low accuracy for nonlinear test dataset (~50%).This agrees with the fact that perceptron being a linear classifier cannot efficiently classify a non-linear dataset.

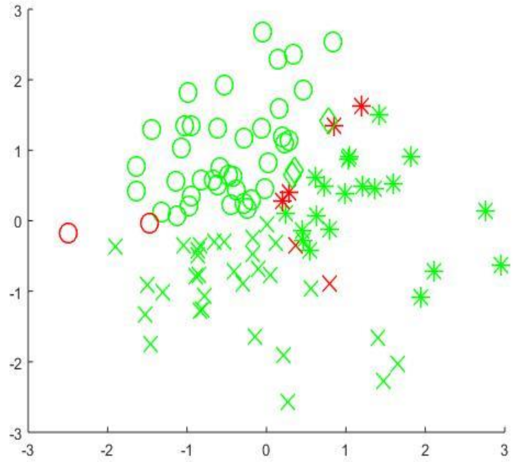


Figure 3: Perceptron Results for Linear Test set

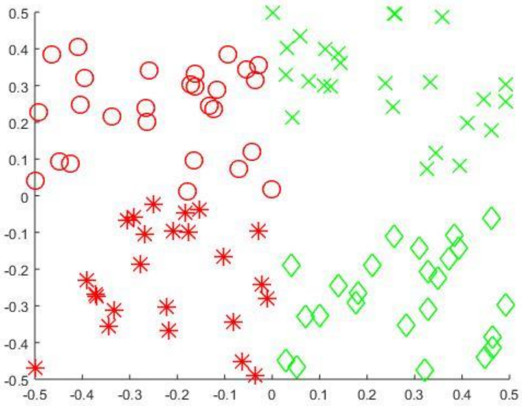


Figure 4: Perceptron Results for Non Linear Dataset

3) logTrain.m-returns an optimum model which is optimized by minimizing NLL using gradient descent
logRegNLL.m- returns the NLL and the gradient value for the given dataset.
logRegTest.m- trains the logistic regression function for different values of lambda and plots the accuracy by evaluating the resulting model on the test dataset.

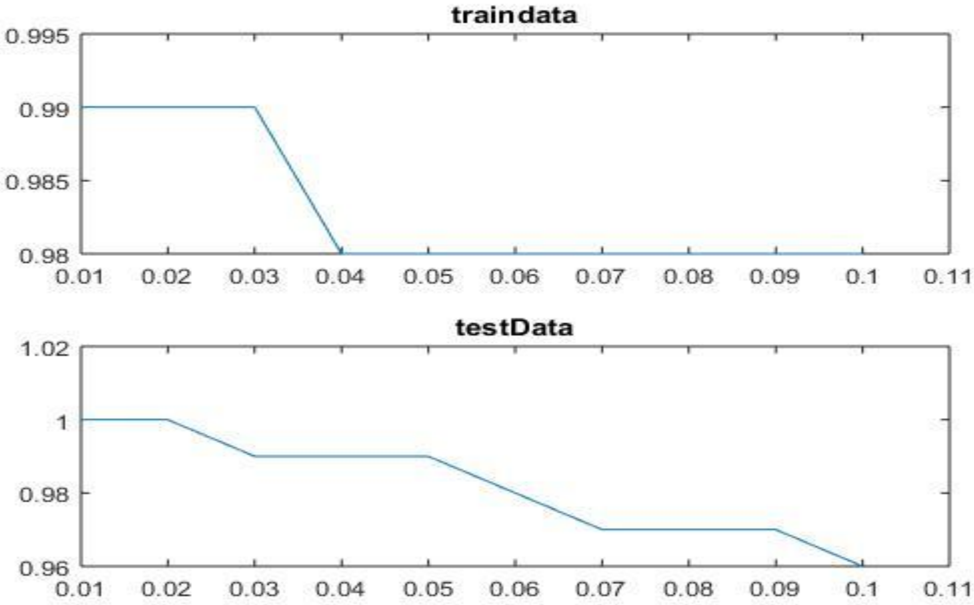


Figure 5: Accuracy of Logistic Regression(y - axis) with variation in lambda for Linear dataset

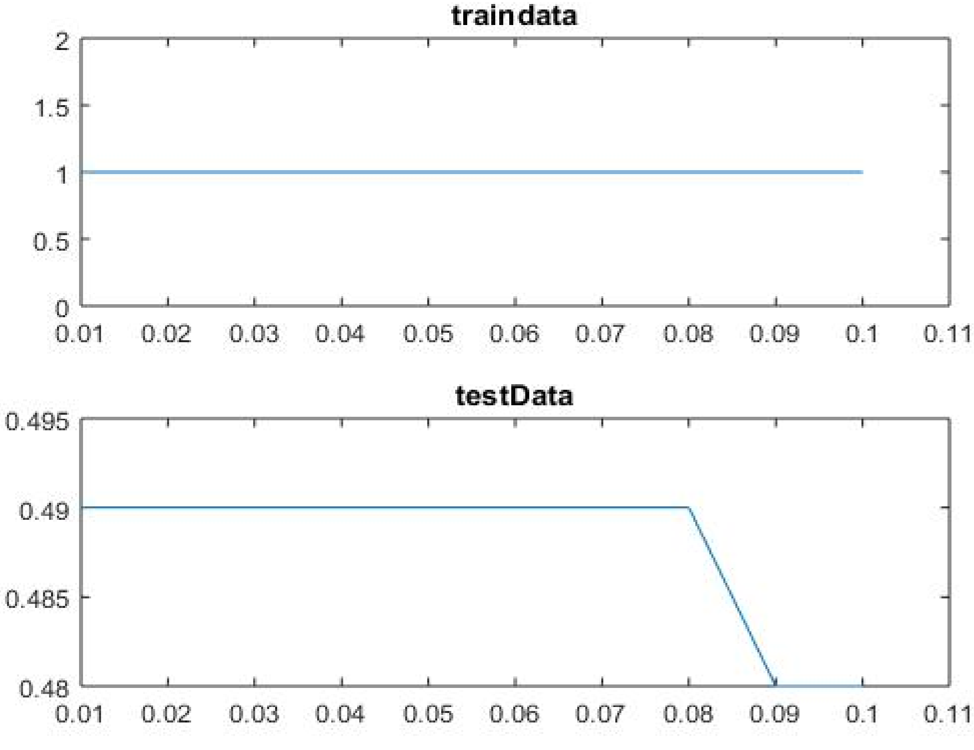


Figure 6: Accuracy of linear regression (Y-axis) for Non-Linear dataset with variation in lambda.

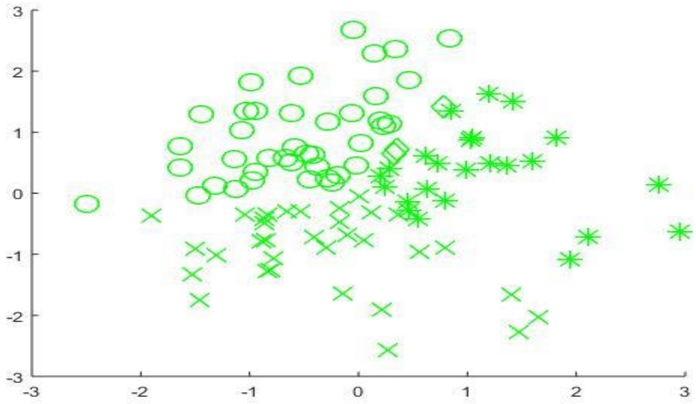


Figure 7: Logistic Regression on linear dataset

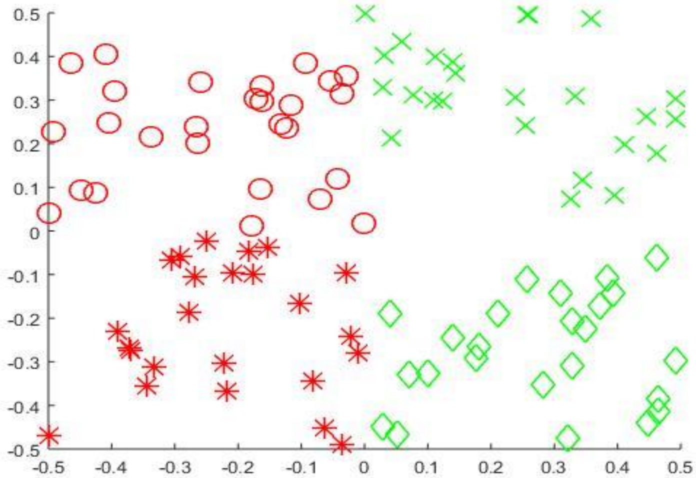


Figure 8: Logistic Regression on nonlinear Dataset

4)

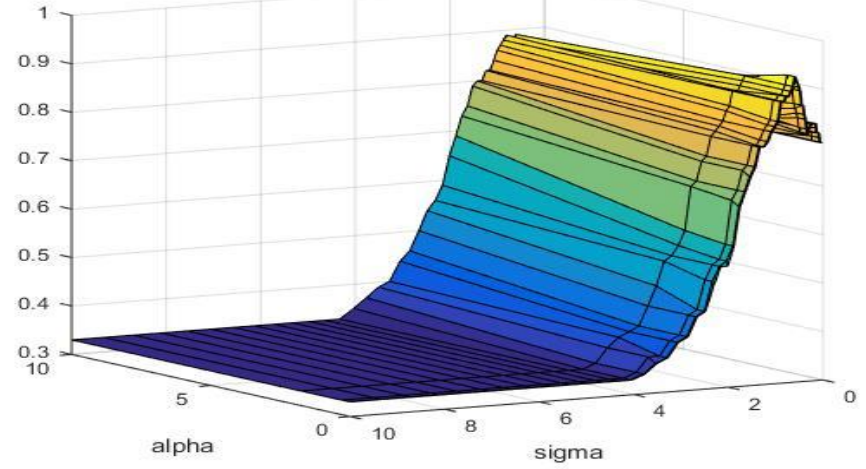


Figure 9: Gaussian Bayes accuracy with parameter sweep (Linear Data)

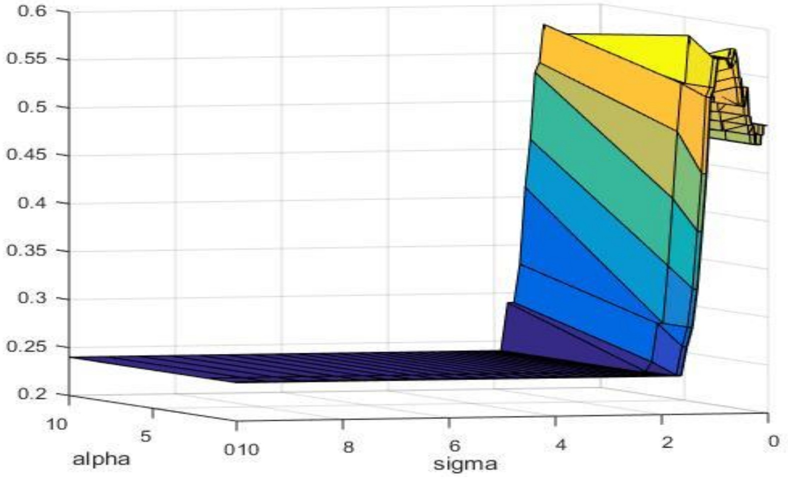
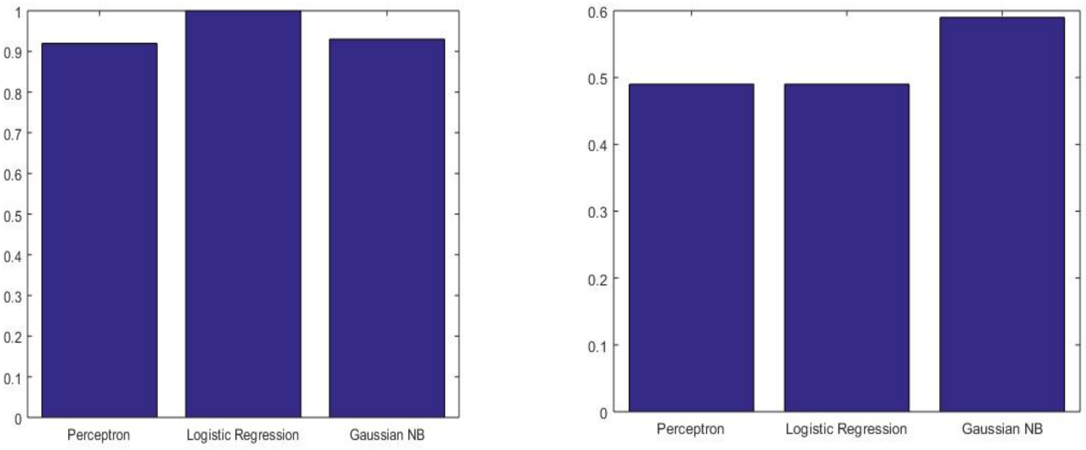


Figure 10: Gaussian Bayes Accuracy with w.r.t parameters (Non Linear Data)



Linear Data

Non Linear Data

Figure 11: Accuracy of all models over the Synthetic Data Sets.

6) Analysis on Cardio Dataset

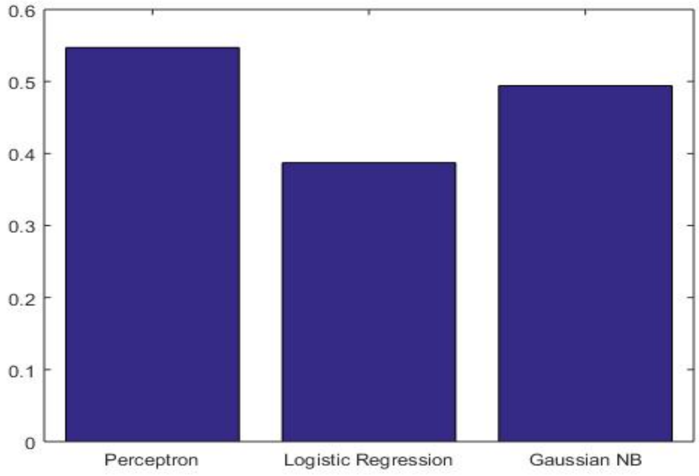


Figure12: Accuracy over cardio Dataset

Perceptron performs the best for the given dataset

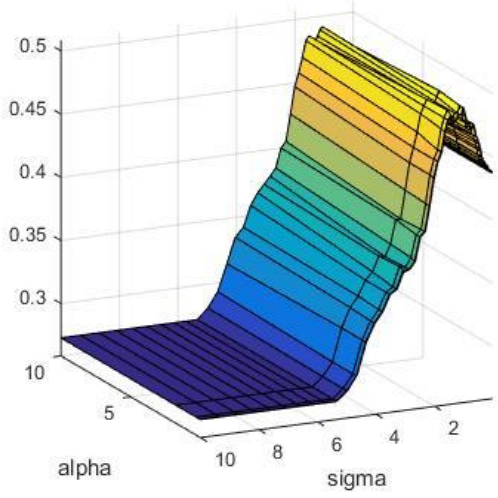


Figure 13: Parameter Sweep Gaussian Naïve bayes

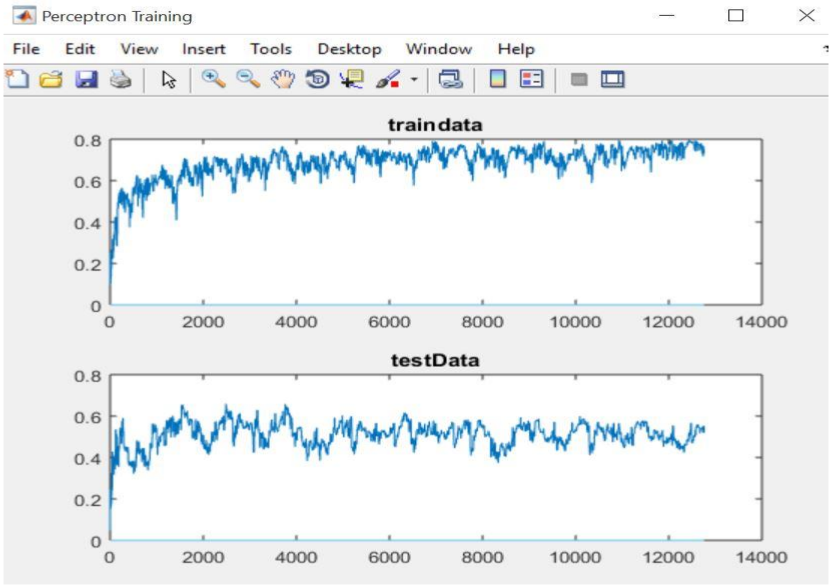


Figure 14: Perceptron Training

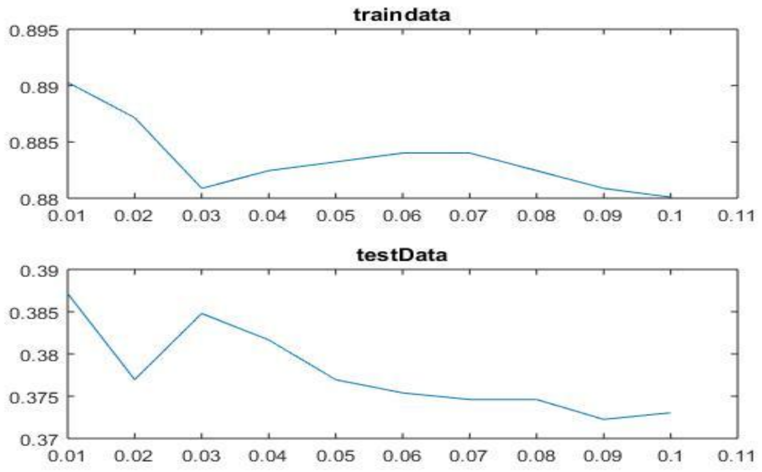


Figure 15: Lambda Sweep-Logistic Regression

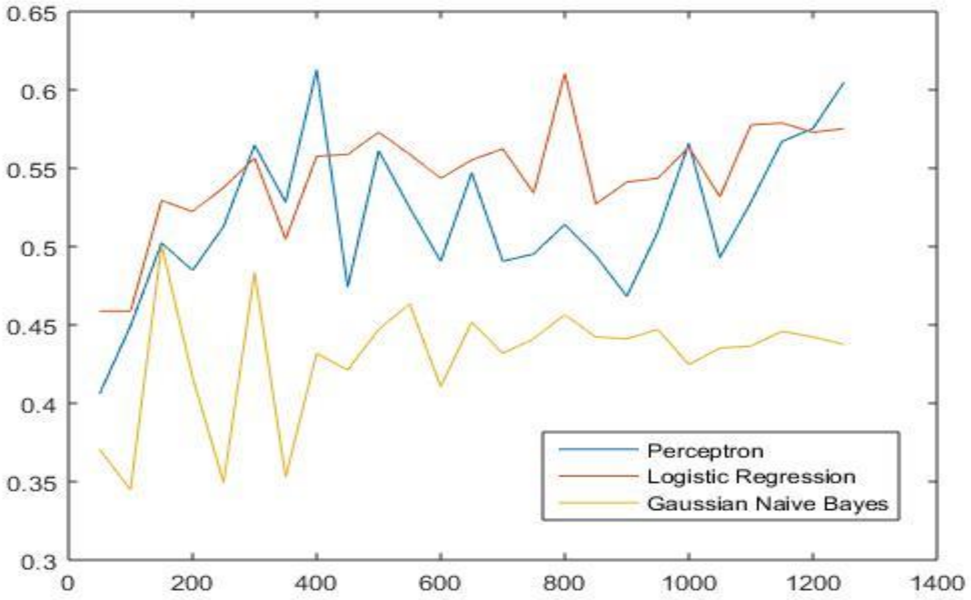


Figure 16: Compare different methods with size of train Data on the x-axis.

CompareAllMethods.m randomly takes datasets from all the given Cardio data with the size of dataset increasing by 50 in each iteration (fig. 16). The perceptron performs the best for the given dataset while the naïve Bayes performs the worst. In general, we expect that generative naïve Bayes to initially do better, but as the sample size is increased, logistic regression to catch up and give higher accuracy on for larger datasets [reference 1].Here, both methods perform similarly upto sample size of 200, after which logistic regression performs better.

Note that the results are slightly different from figure 12 where the difference being that the data is randomized while in the runCardioExperiments.m, the data was not randomized. The most interesting observation is in the performance of Logistic regression which increases significantly over randomly selected dataset.

The perceptron does nearly as well as the logistic regression because of the inherent similarity in the model (both try to optimize W, with the objective being some function of $\mathbf{w}^T.\mathbf{x}$ ) the difference being perceptron does online learning while logistic regression doing batch learning.

References:

[1]Andrew NG, Michael Jordan,"On Discriminative vs Generative Classifiers: A comparison of Logistic Regression and naïve Bayes".