CSC 242 Project 4

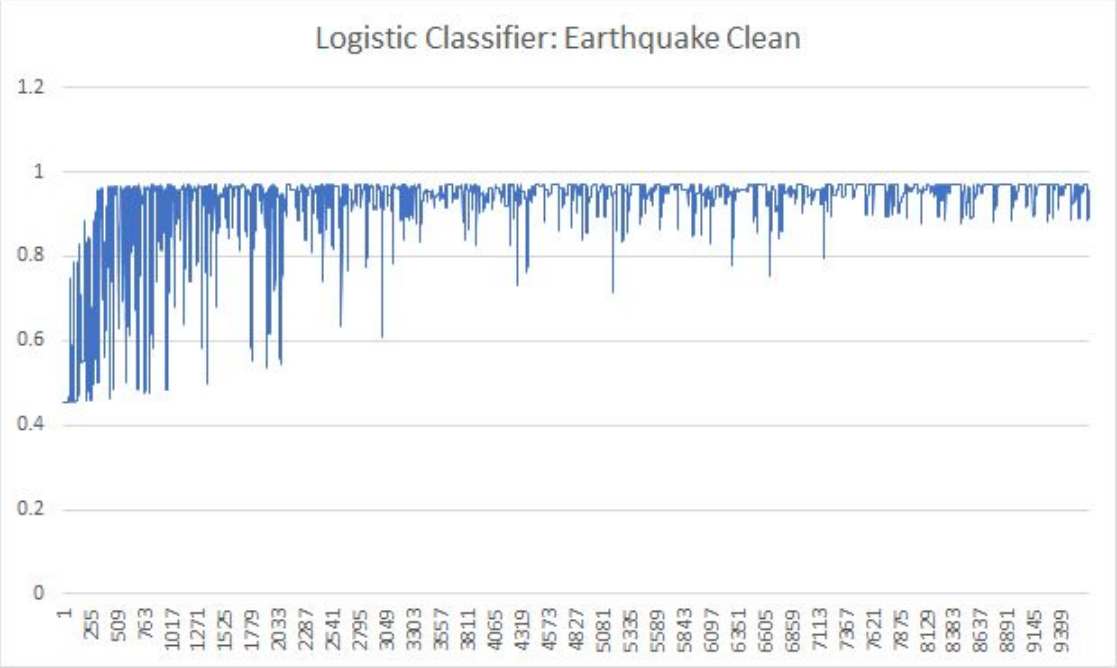
Arun Ramesh and Thabani Dube

Main method:
proj4Main.java for classifier
MainIris.java for neural networks (both in default package).

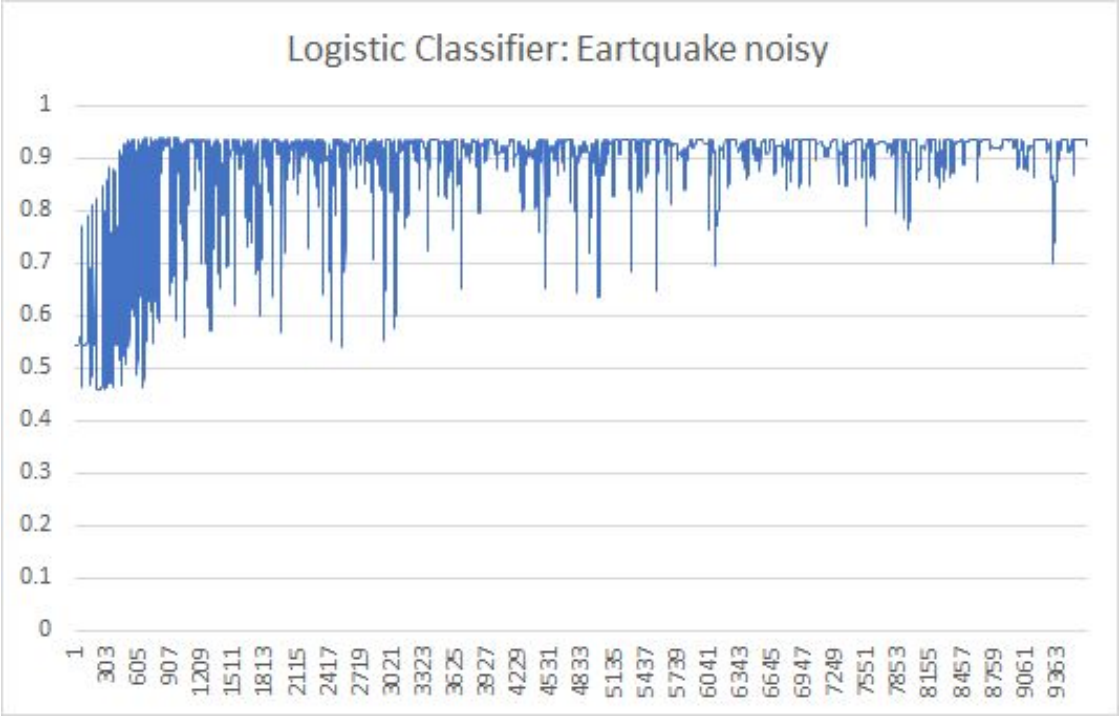
Run configurations:
[logistic/perceptron] [filename] [numberOfSteps] [learning Rate]
If decaying learning rate, set learning Rate = 0

Logistic Classifier:

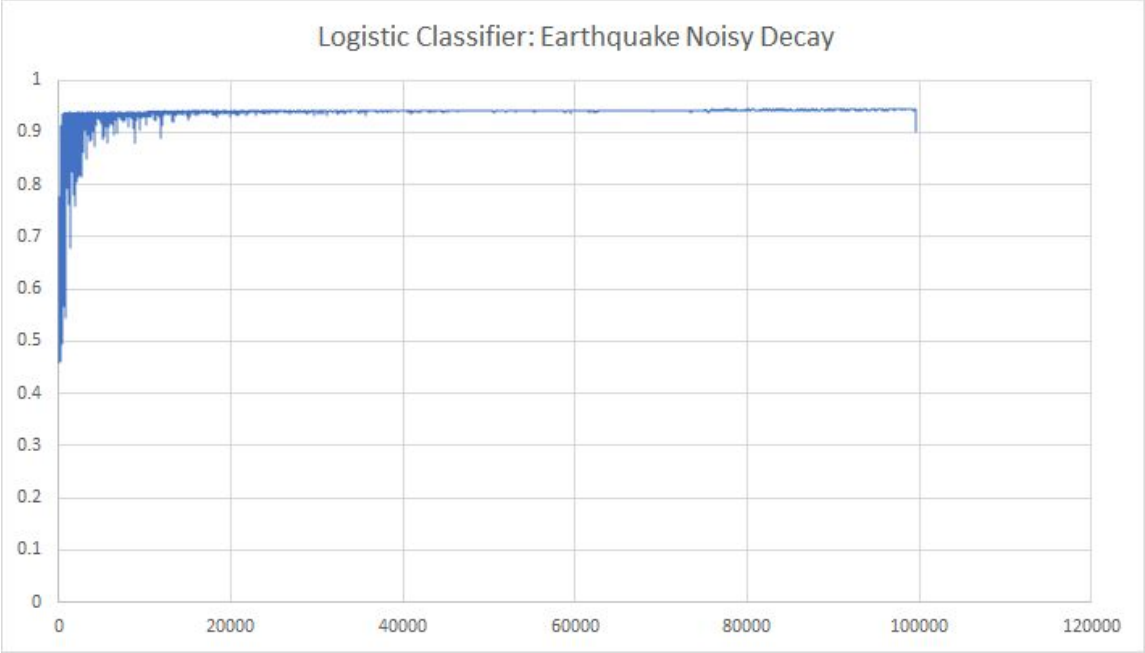
Run configurations: logistic earthquake-clean.data.txt 10000 0.95



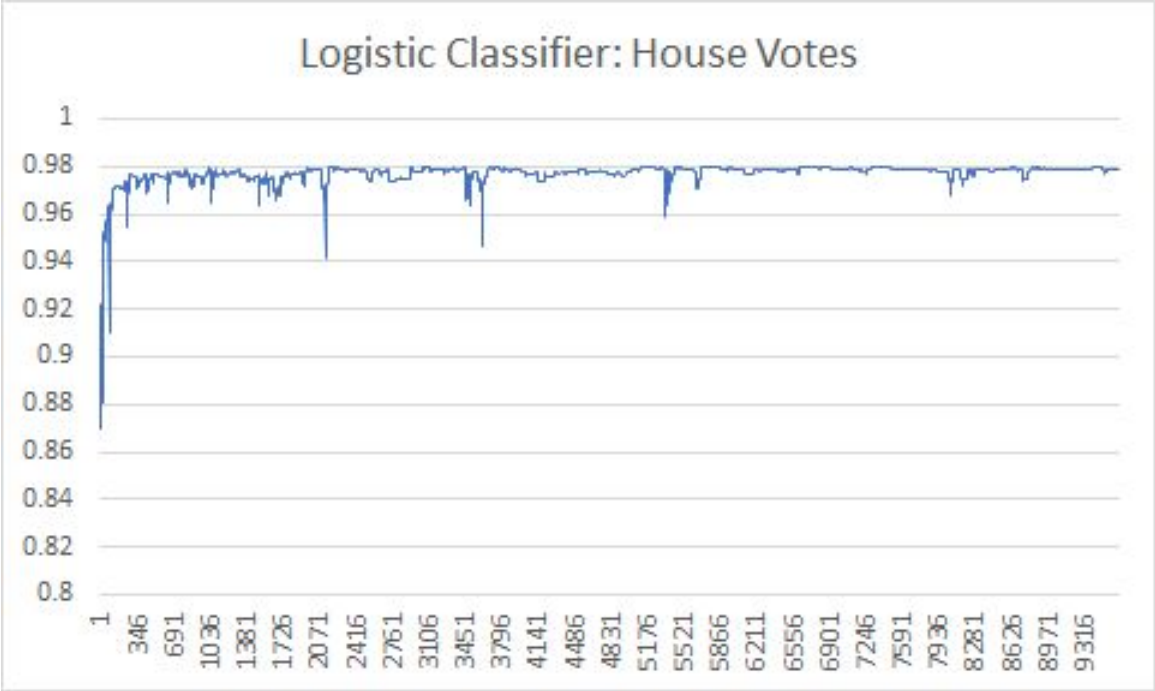
Run configurations: logistic earthquake-noisy.data.txt 10000 0.95



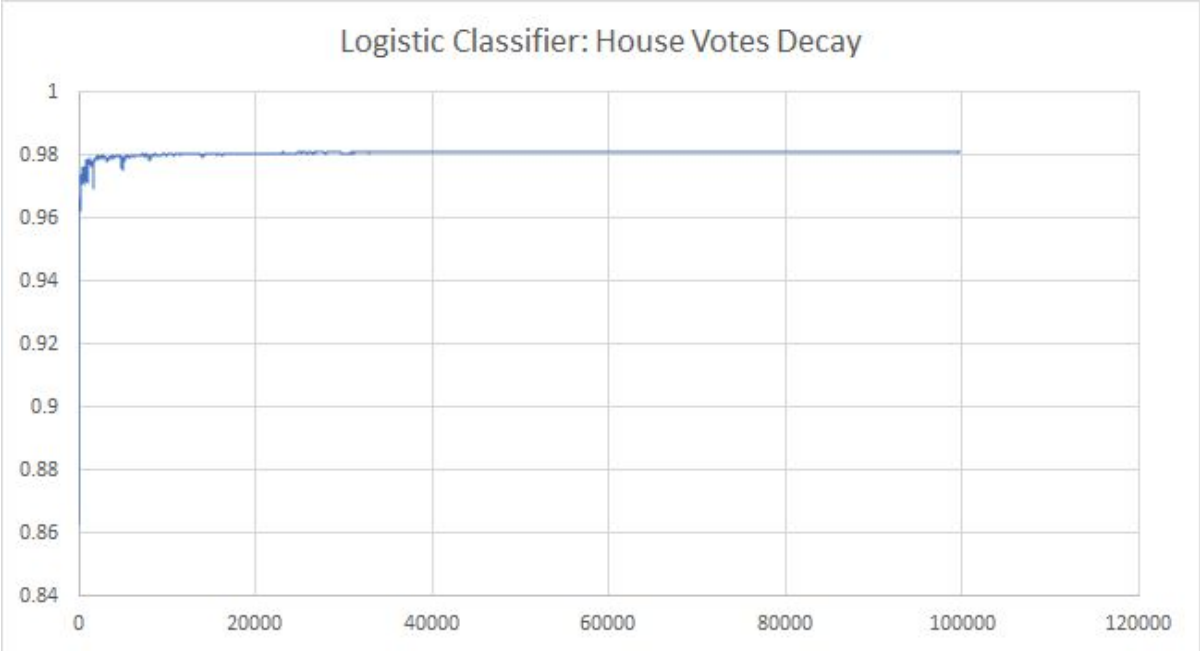
Run configurations: logistic earthquake-noisy.data.txt 10000 0



Run configurations: logistic house-votes-84.data.num.txt 10000 0.95

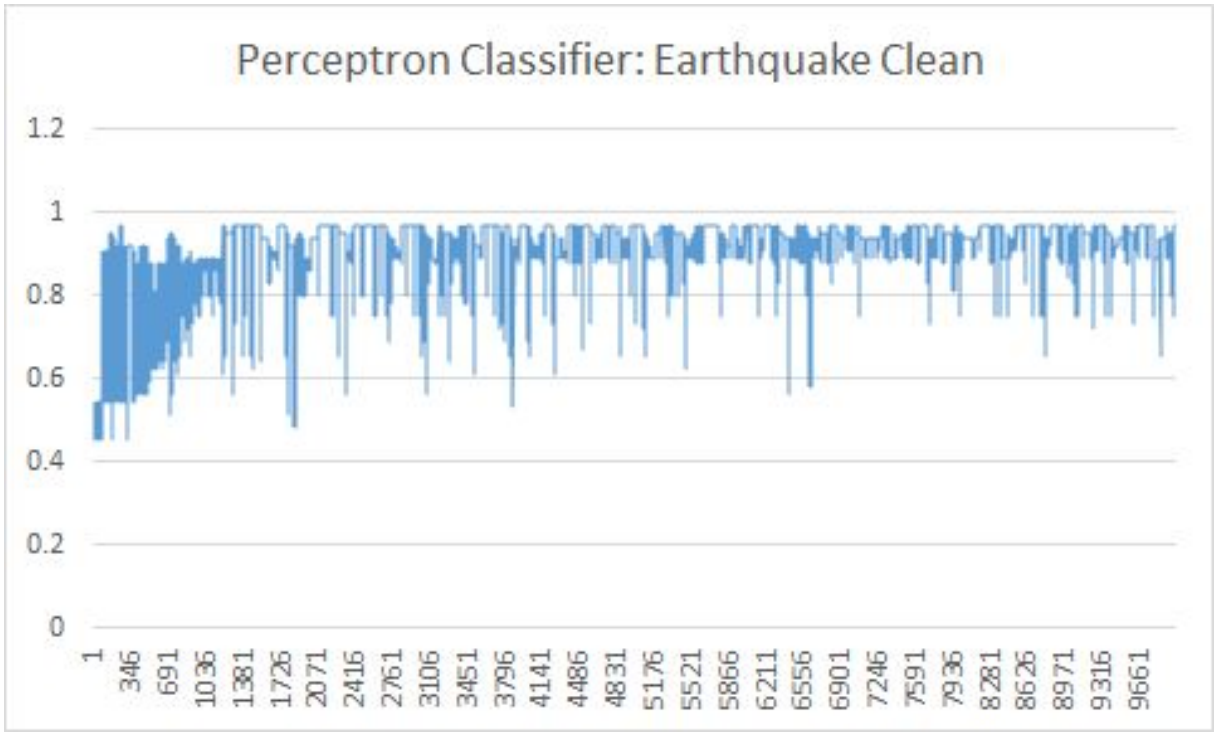


Run configurations: logistic house-votes-84.data.num.txt 10000 0

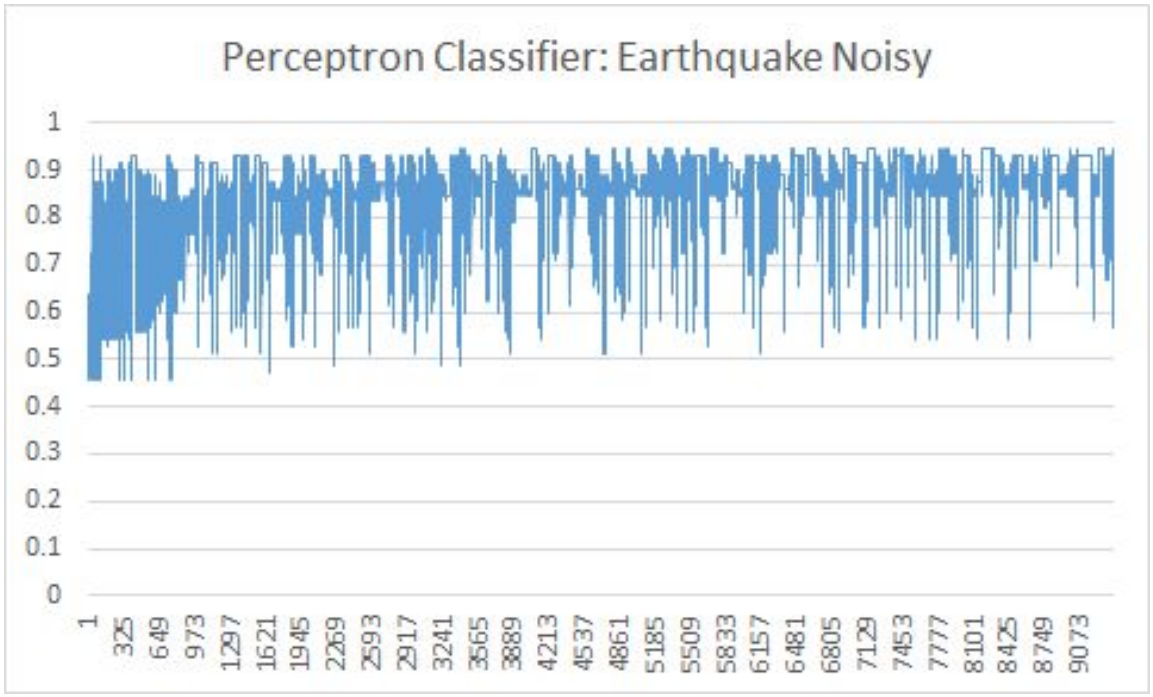


Perceptron Classifier:

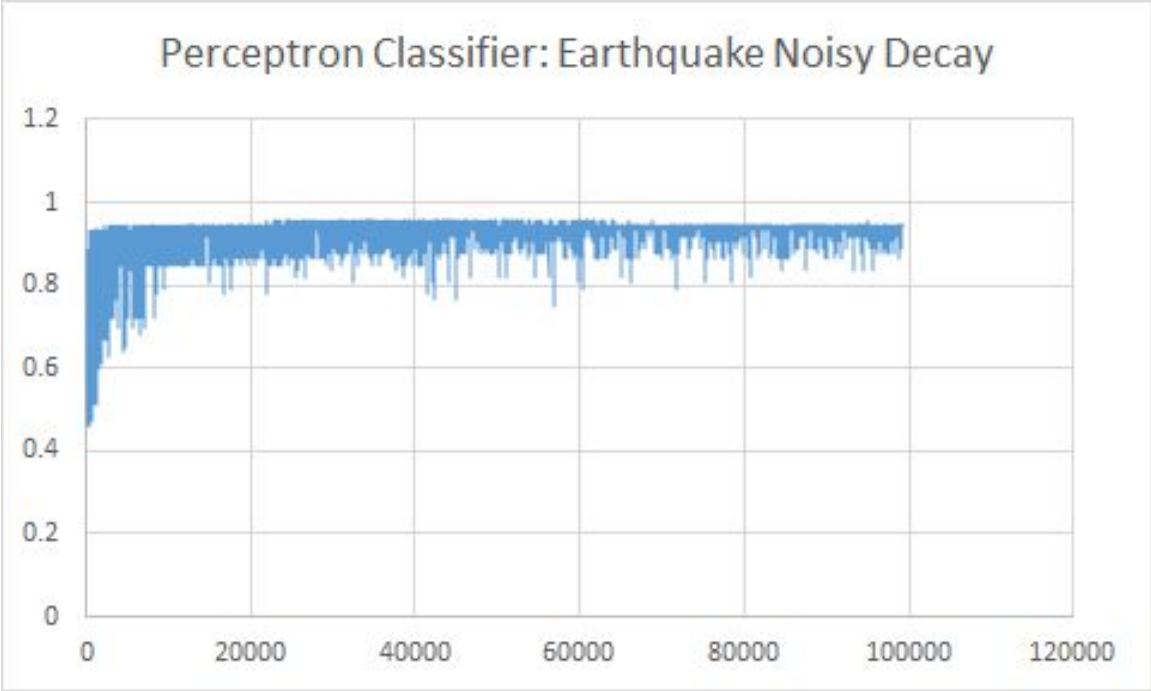
Run configurations: perceptron earthquake-clean.data.txt 10000 0.95



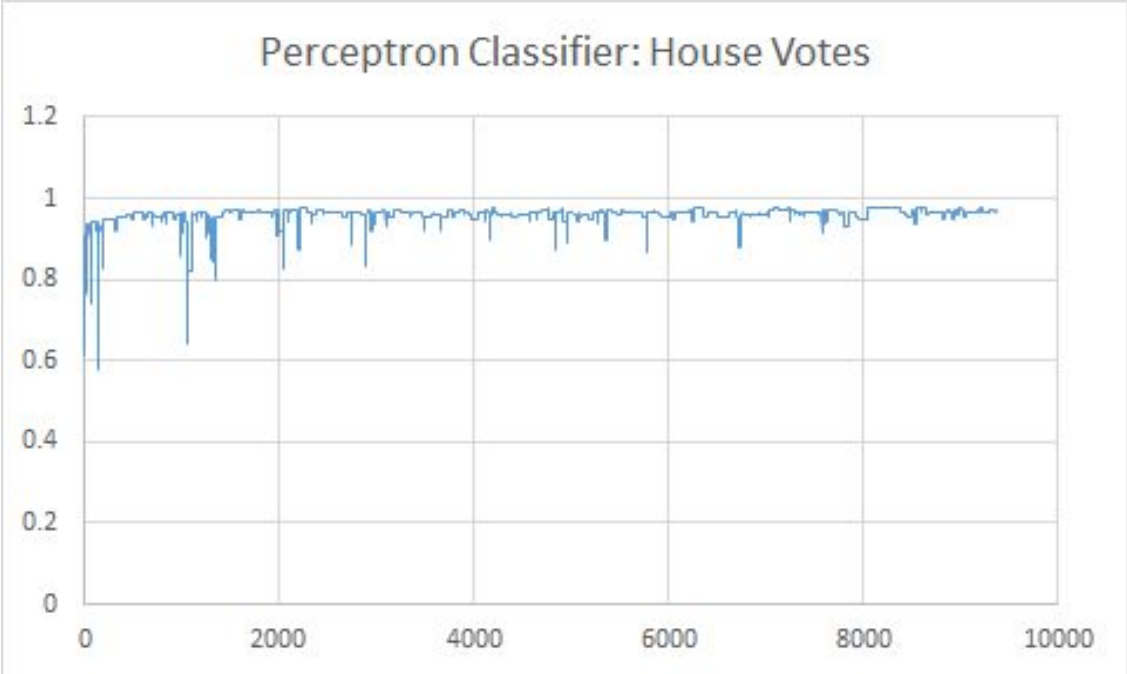
Run configurations: perceptron earthquake-noisy.data.txt 10000 0.95



Run configurations: perceptron earthquake-clean.data.txt 10000 0



Run configurations: perceptron house-votes-84.data.num.txt 10000 0.95



NEURAL NETWORKS

We have implemented neural networks for iris and mnist datasets.

Refer to class MainIris.java which contains main method to run program
The networks have good accuracy (please refer to suggested run configuration), and you can see that they increase with an increase in the number of epochs.

IRIS dataset

Run configurations:

[iris] [nepochs] [learning rate]

Eg

iris 1000 0.1

MNIST dataset

Run configurations:

[mnist] [nepochs] [learning rate]

Eg

mnist 100 0.1

Mnist takes quite a while to run, as we have to convert the images as well.