Ryerson University
Department of Electrical
& Computer Engineering
Faculty of Engineering & Architectural Science

Course Title:	**Intelligent Systems**
Course Number:	**ELE 888**
Semester/Year (e.g.F2016)	**W2021**

Instructor:	**Dr. Xiao-Ping Zhang**

Assignment/Lab Number:	**Lab 2**
Assignment/Lab Title:	**Linear Discriminant Functions**

Submission Date:	**Feb-28-2021**
Due Date:	**Feb-28-2021**

Student LAST Name	Student FIRST Name	Student Number	Section	Signature*
An	Nicholas	500750378	3	NA
Parmar	Gurvir	500765786	1	GP
Shreekant	Vatsal	500771363	1	VS

**By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a "0" on the work, an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:* http://www.ryerson.ca/senate/current/pol60.pdf

Objective:

To find a linear combination of features which characterizes or separates two or more classes of objects or events, a linear discriminant function is used. This function is used in statistics, pattern recognition, and machine learning. These functions are linear in either the component of the feature vector x or in some given set of function of x.

A linear discriminant function g(x) can be written as:

$$g(x) = w^T x + w_0 \tag{1}$$

where w is the weight vector and w_0 is the threshold. If there are two categories we decide w_1 by observing the feature vector x if $g(x) > 0$ or w_2 if $g(x) < 0$.

A generalized discriminant function can be represented as:

$$g(x) = a^T y \tag{2}$$

where y is the augmented feature vector and a is the augmented weight vector.

The objective of this lab was to classify the Iris datasets obtained from "archive.ics.uci.edu/ml/" using linear discriminant functions and to compute the weight vector using the training data samples. The perceptron criterion function $J_p(a)$ and its respective gradient ∇J_p can be represented by:

$$J_p (a) = \sum_{y \in Y} (-a^T y) \tag{3}$$

$$\nabla J_p = \sum_{y \in Y} (-y) \tag{4}$$

Observation:

Class A and Class B

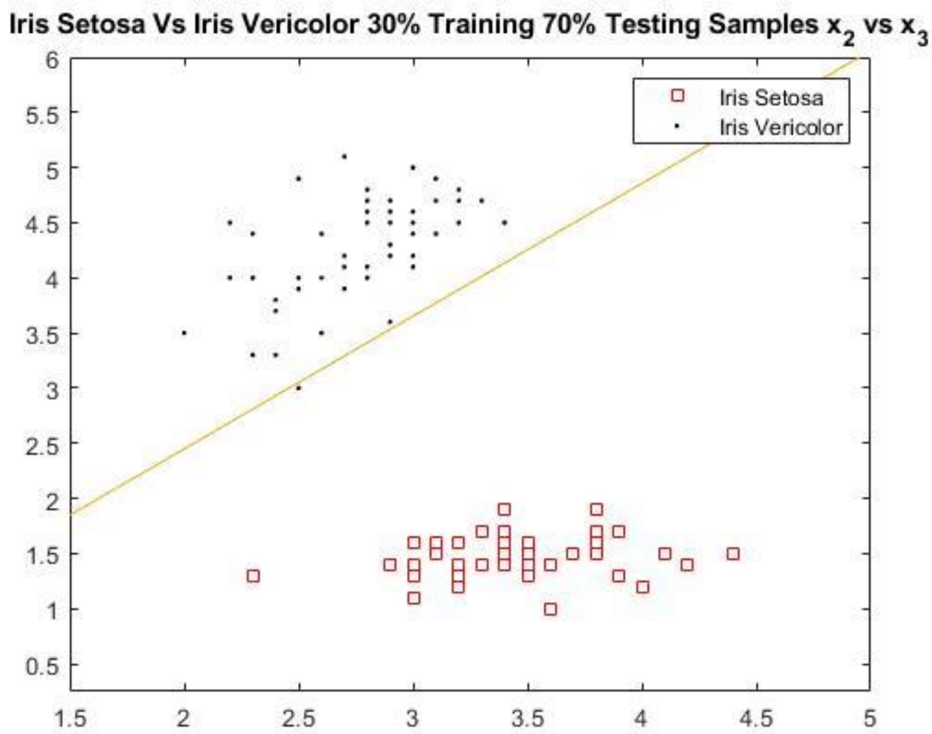

Figure 1: Data split with 30% training

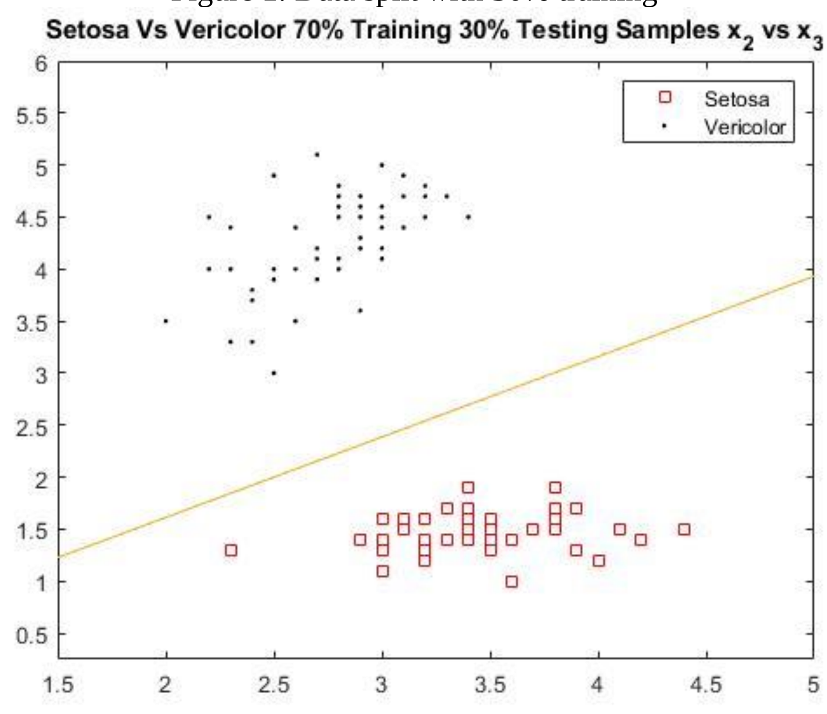

Figure 2: Data Split with 70% training

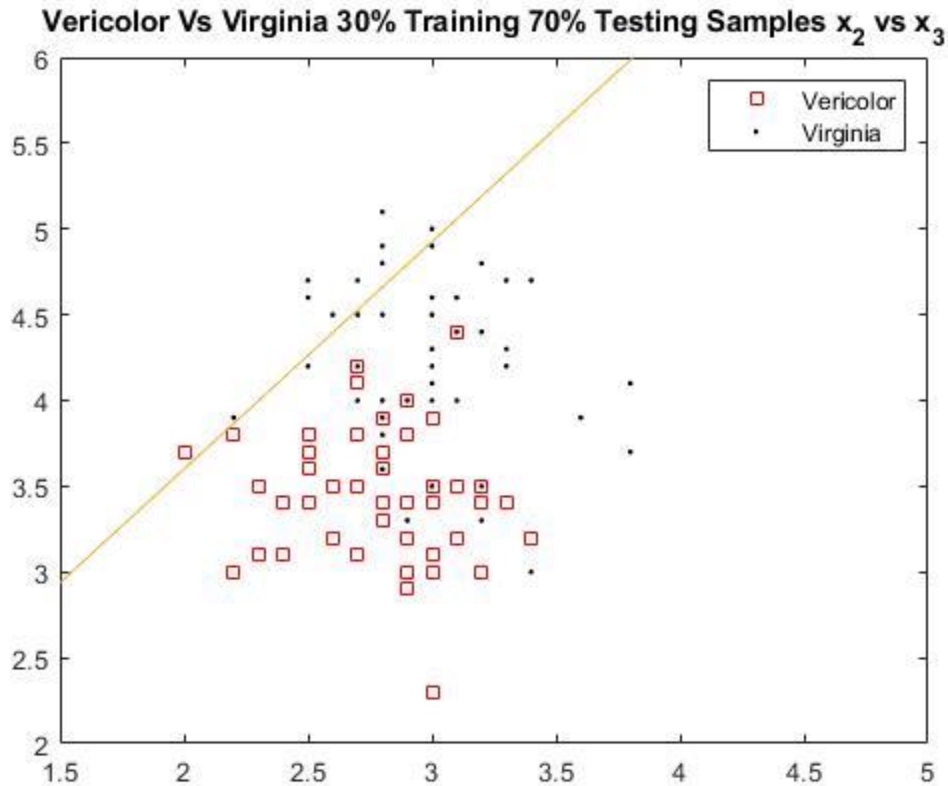

Figure 3: 30% training with decision boundary

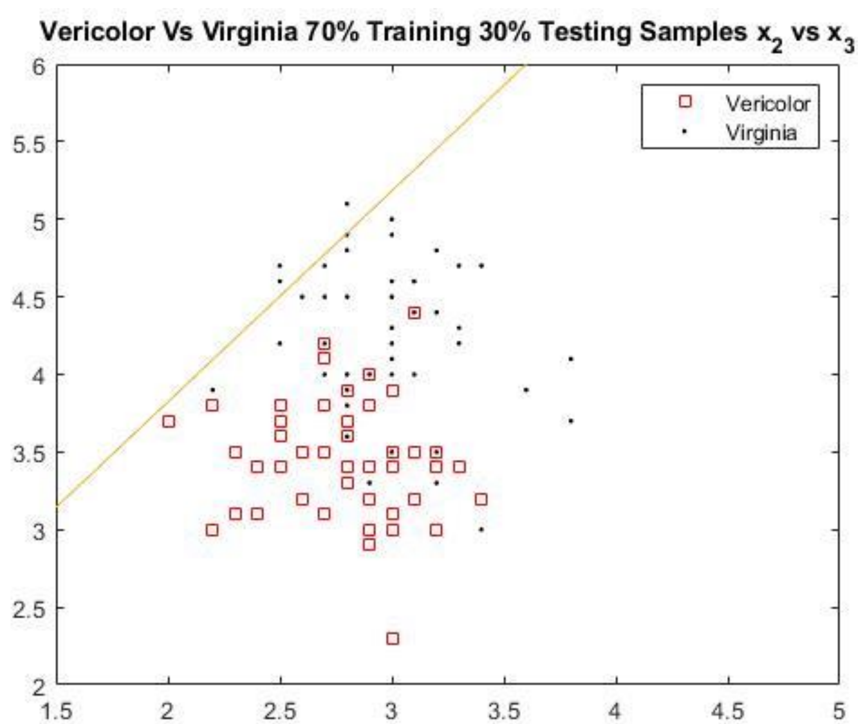

Figure 4: 70% training with decision boundary

Numerical Values:

Accuracy is:

Class A	Class B	Training Set Percentage
98.571	100	30
84.286	80	70

Class B and Class C

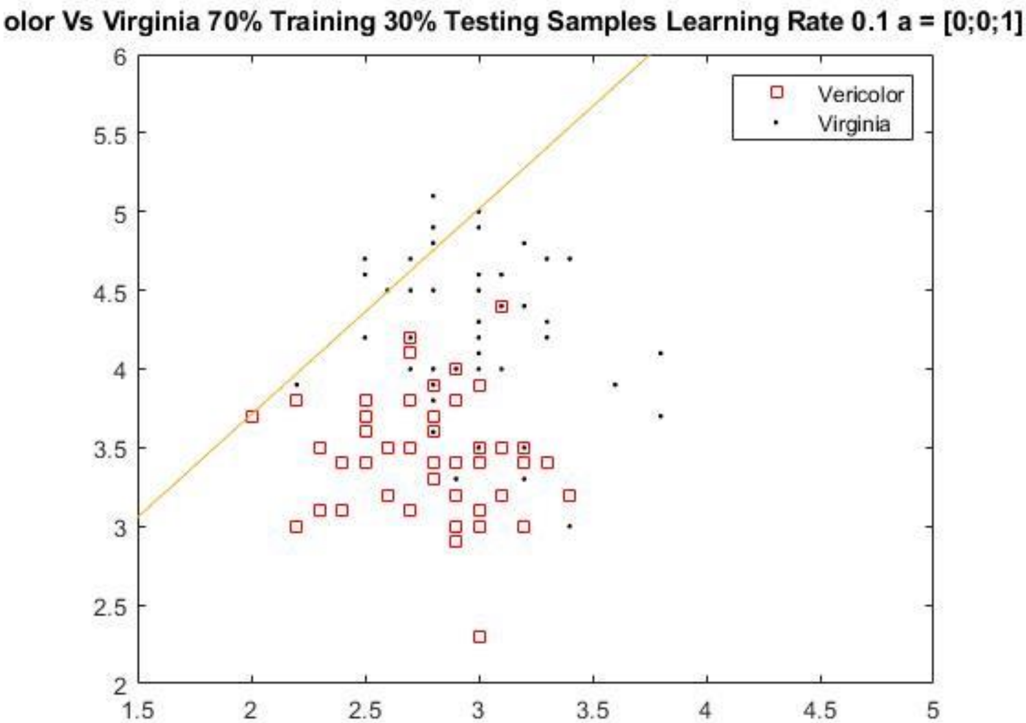

Figure 5: Data split with 30% training

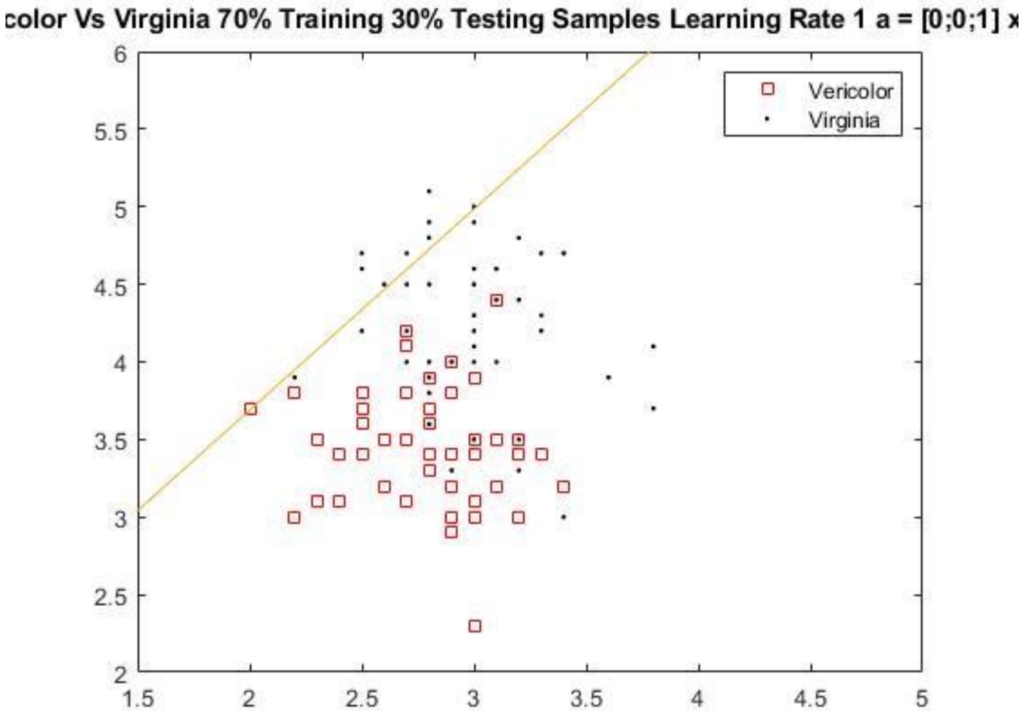

Figure 6: Data split with 70% training

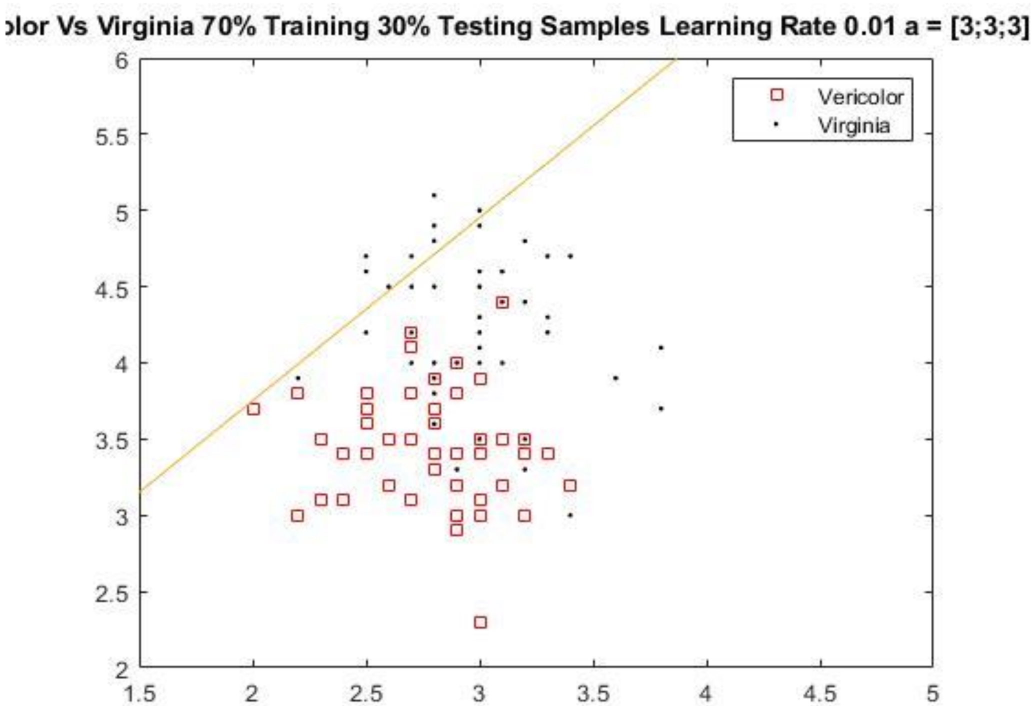

Figure 7: 30% training with decision boundary

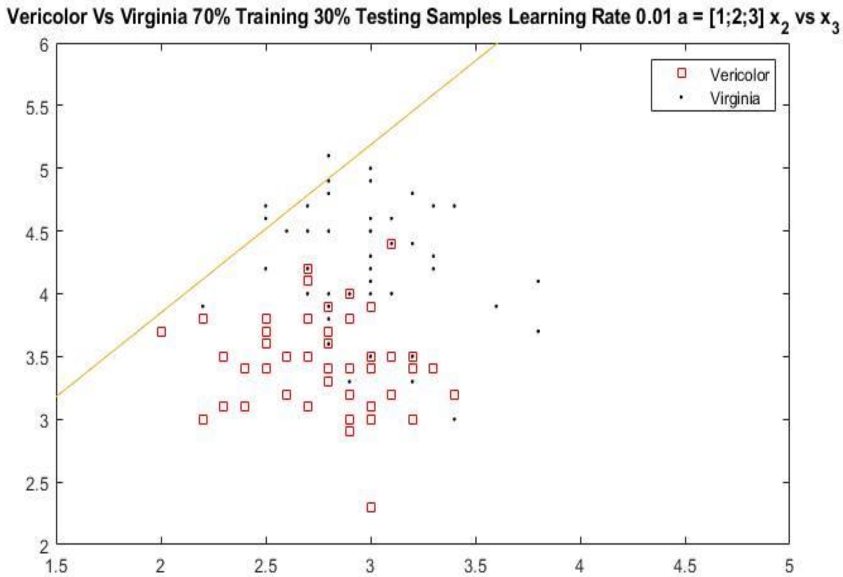

Figure 8: 70% training with decision boundary

Numerical Values:

Accuracy:

Class B	Class C	Training Set Percentage
86.667	86.667	30
90	80	70

Discussion:

i. <u>The effect of using different sizes for training and test data:</u>

By utilizing a larger training set the gradient descent will converge in less iterations since it has more data to pass through. This is evident if you observe the time it took the AB training dataset to converge compared to the AB training dataset which is twice as large as its counterpart.

ii. <u>The effect of different learning rates, threshold, initial weight values:</u>

Different Learning Rates:

By utilizing a finer learning rate, you increase the time required to converge and sometimes it is possible to over fit the dataset thus producing error. By utilizing a coarser learning rate, you decrease the time required to converge and again you can under fit the dataset producing error.

Threshold:

The threshold is essentially the acceptable amount of error, so by changing the threshold you are allowing a certain amount of error. Therefore, it is ideal to keep the error as low as possible, but with very large datasets this may not be possible due to the computation time required.

Initial Weight Values:

By changing the initial weight values, it can have many effects, for instance, it allows you produce zero error and converge sooner. However, sometimes you may converge sooner, but there is more error.

iii. The criterion function over the iterations:

Since the criterion function is based the on the features that were miss classified it behaves like a bang-bang control system, where it will correct itself over time by overshooting, then undershooting until it converges correctly.

iv. The classification accuracy for the given data:

For the dataset AB, a perfect classification was achieved based on the initial values when both the training and testing sets where used to train. When using dataset BC, the features were similar where the error was close to 50%. This occurred since everything was being classified as class 2, had the amount of allowable iterations been increased convergence may have been possible.

Conclusion

Utilization of different sizes of data led to drastic changes when they were used for testing and training. With training data being a higher value, more accurate criterion function for the set and the usage of more testing data resulted in determining the accuracy of the criterion function. Based on the graphs, the accuracy can be tweaked by changing the size of the training set. It was noted that larger training sets lead to different accuracies than smaller training sets. Additionally, the perceptron discriminant function can also have an impact on the learning rate. It was observed that with an increasing learning rate, lesser iterations were required for the results to converge.