Silas Rodriguez

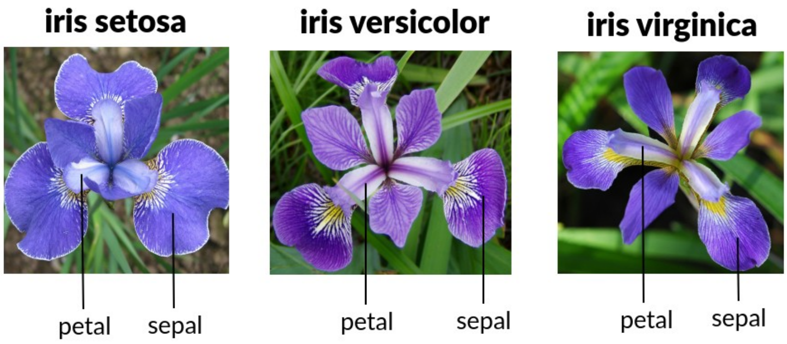
ECE 5363

Pattern Recognition

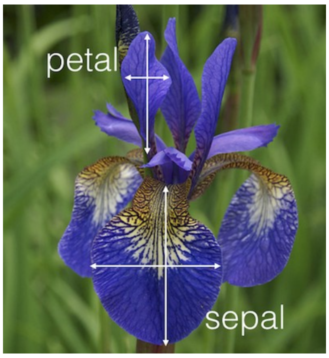
February 7, 2024

ECE 5363 Project 1 Report

1) Loading the Dataset:
   a) There are 3 classes, M = 3
   b) There are 4 Observable Features, l = 4
   c) I researched the iris flowers online and happened across several images in particular that helped me gain insight into the measurements.
      - This graphic showed me what the 3 classes of flowers looked like, as well as pointed out explicitly what it was I was observing on each. Setosa's appear to have wider sepals but not by much. And the Versicolor & Virginica are hard to distinguish. From this image alone, I would have thought to measure the color / shade of blue. But I could see the length of a petal performing particularly well if iris flowers have a large variance between classes.



      - This graphic helped distinguish what a petal was vs a sepal from a more observable angle. Using the graphic above, I was assuming this would be a setosa or virginica. I assume so based on the color (darker blue) and the sepals being the oval / longer shape suggest to me that I could logically narrow my guess to virginica.



2) Computing the statistical quantities for the features

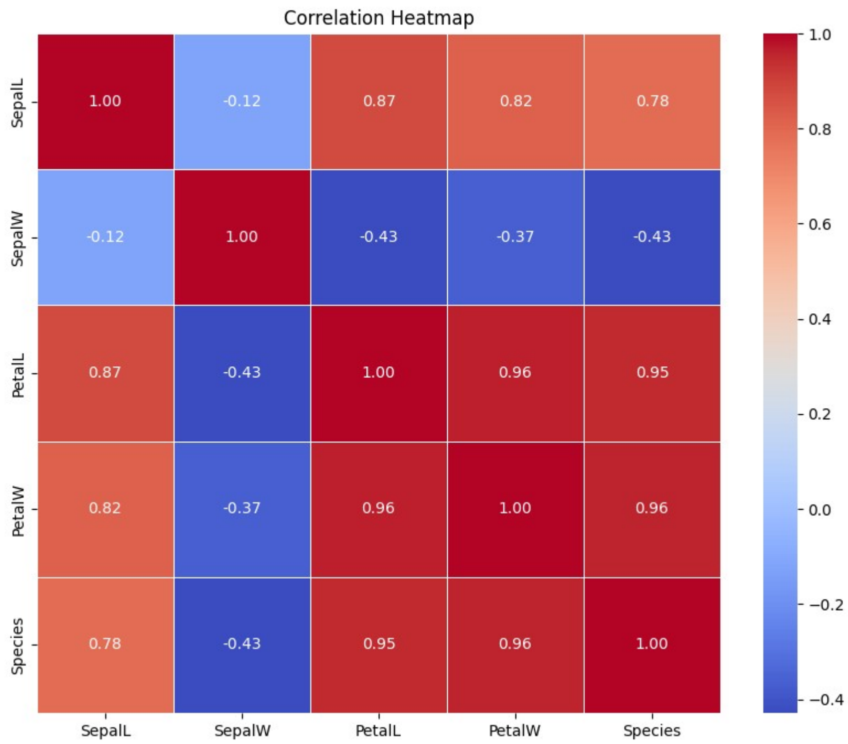
	Sepal L	**Sepal Width**	**Pedal Length**	**Pedal Width**
Minimum	4.3	2.0	1.0	0.1
Maximum	7.9	4.4	6.9	2.5
Mean	5.843	3.057	3.758	1.199
Variance	0.681	0.189	3.096	0.577
Within-Class Var	0.260	0.113	0.182	0.041
Between-Class Var	0.421	0.076	2.914	0.536

a) I computed these using numpy's library functions with exceptions to the within-class variance and between-class variance.

b) I notice pedal width has smaller values compared to other classes. Additionally, the between class variance for Pedal length is the highest, suggesting the spread between classes using this feature would be good. It also has the 2nd largest within-class variance, so that suggests that within a class, these data points are slightly less packed together. Within a class, petal width observes the most packed features measurements.

c) The data would likely benefit from standardization in order to better discriminate features due to varying means, and large variance among points within features.

3) Computing the correlation coefficient matrix & analyze the display



a)

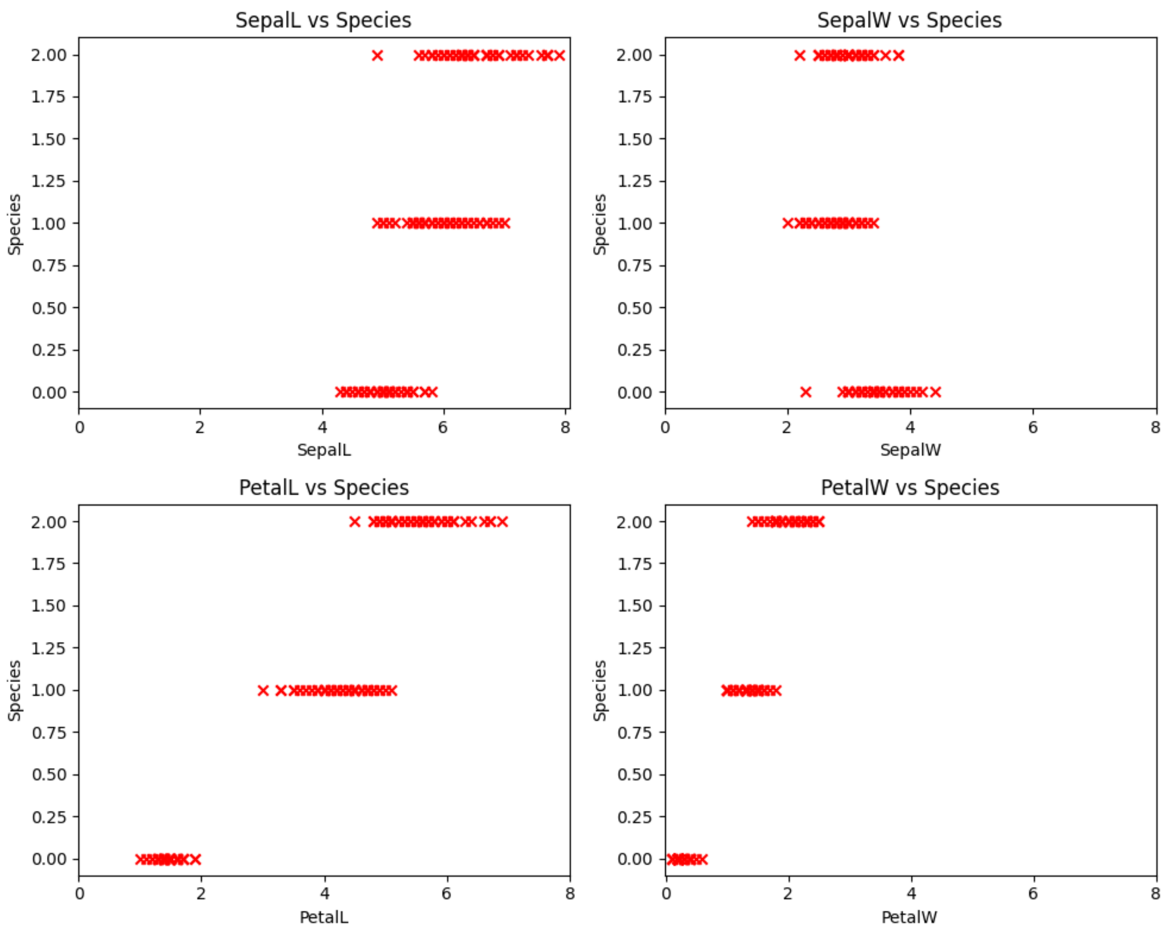
b) I find the most interesting place to start is the features. For example, strong positive correlations between petal length & all the other features besides sepal width. This leaves me to believe this feature would not be easily linearly separable since it would scale with the others positively and strongly.

c) What else is interesting is that the correlation between sepal length and width is weak, suggesting these may be two features worth classifying over because a change in one feature doesn't lend itself to changing the other linearly.

d) However, in the species column, I can see the Petal Length and Width are most correlated with the classification. Strong positive correlation so that must mean that higher classes are associated with longer lengths and widths, similar to the photo I mentioned in part 1.C.

4) Display the features vs. class labels on an axis and predict the performance of classification

a)



b) I see a lot of horizontal overlap between classes in the Sepal L and Sepal W features. This suggests these would not be easily linearly separable, if at all. I would not use these to perform classification due to increased complexity, time, and likely poorer performance at generalizing new data points. Perceptron won't converge.

c) On the other hand, petal length displays better separation of the data, and suggests it would converge with Perceptron with class 0 vs others. Petal Width is similar, but not as spread. It would be much easier to linearly classify from these two features and they would more likely converge with Perceptron.

5) For each case classified, report No. Epochs, Weight Vectors, Training, Misclassifications and plot the boundary when necessary. **For w, first term is w0 bias term.**

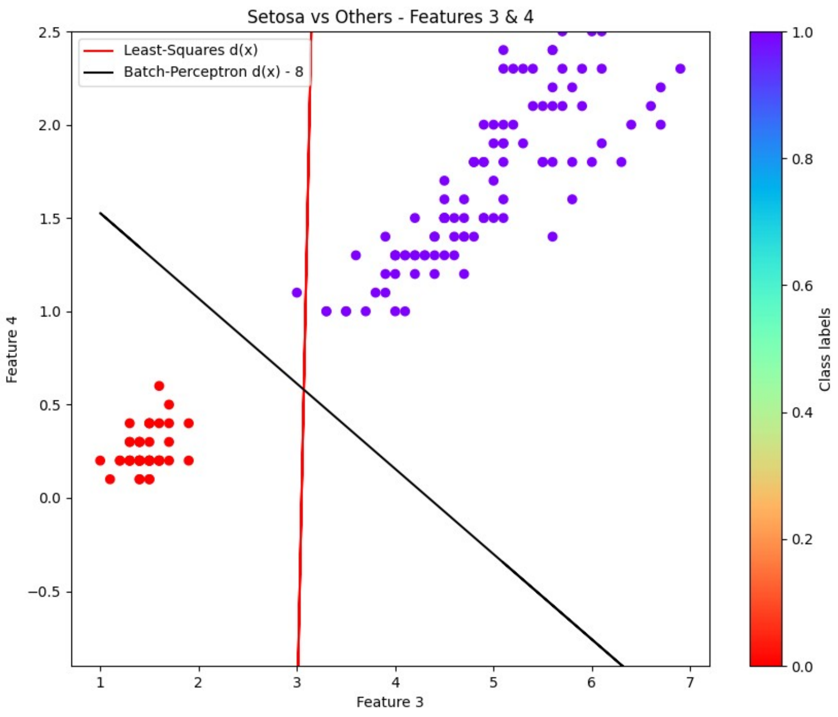
a) Setosa Vs. Versi+Virigi, All Features

	Batch – Perceptron	**LS Closed-Form Solution**
Convergence (BP Only)	Yes, 9 Epochs	
Weights Computed	[8.764 13.670 34.858 -47.419 -21.302]	[0.881 -0.066 -0.242 0.224 0.057]
Misclassifications	0	0
Training Accuracy	100%	100%

b) Setosa Vs. Versi+Virigi, Features 3 & 4

	Batch – Perceptron	**LS Closed-Form Solution**
Convergence (BP Only)	Yes, 8 Epochs	
Weights Computed	[18.923 -4.360 -9.541]	[-0.266 0.251 -0.010]
Misclassifications	0	1
Training Accuracy	100%	99.33%

- Plot:



-

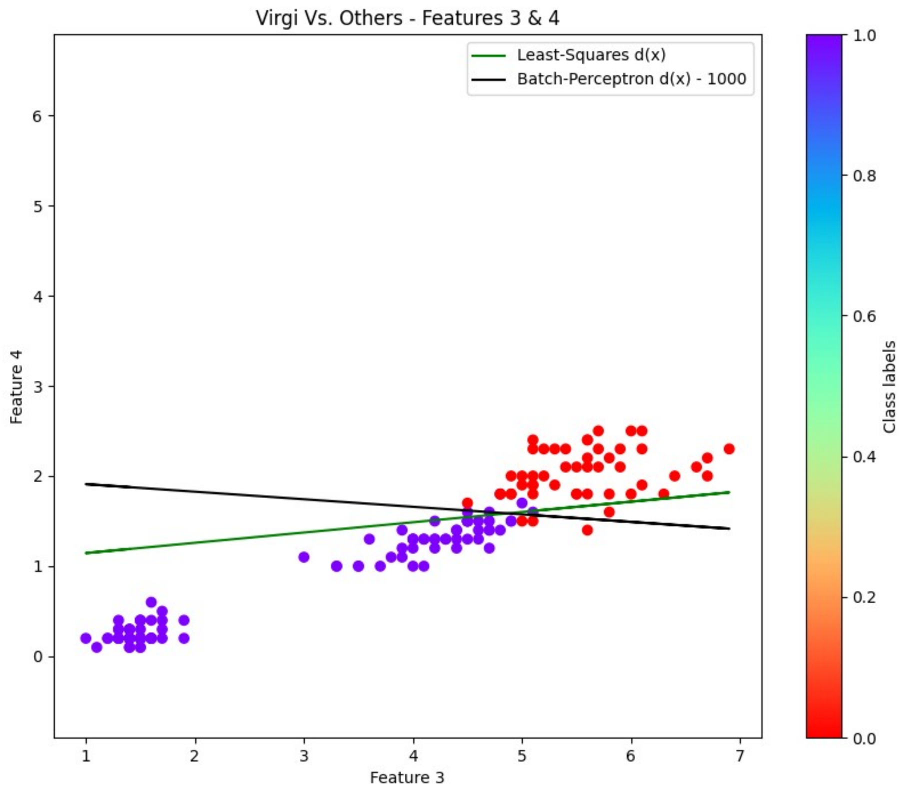
c) Virigi Vs. Versi+Setosa, All Features

	Batch – Perceptron	LS Closed-Form Solution
Convergence (BP Only)	No Convergence, 1000 epochs	
Weights Computed	[-25.732 -55.559 -50.314 82.286 64.647]	[1.695 0.046 -0.203 -0.004 -0.552]
Misclassifications	2	11
Training Accuracy	98.7%	92.7%

d) Virigi Vs. Versi+Setosa, Features 3 & 4

	Batch – Perceptron	LS Closed-Form Solution
Convergence (BP Only)	No Convergence, 1000 epochs	
Weights Computed	[-56.935 2.391 28.561]	[1.160 0.073 -0.640]
Misclassifications	7	8
Training Accuracy	95.3%	94.7%

- Plot:



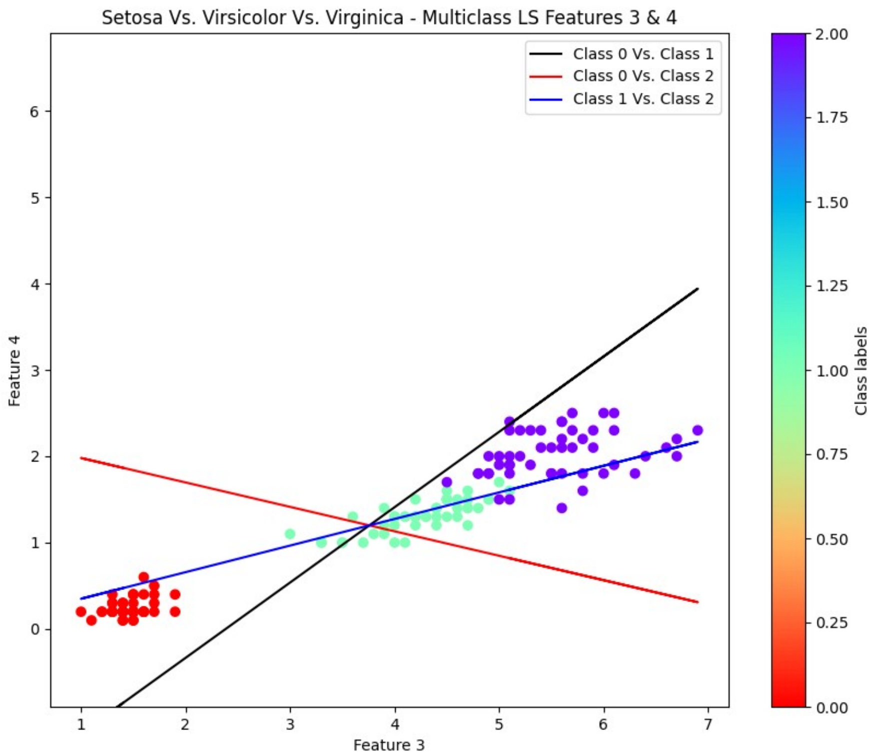
-

e) Setosa Vs. Versi Vs. Virigi. Features 3 & 4 – Multiclass LS

- W – Matrix: (l+1) x M, so d1(x) would be col 1, d2(x) col 2, d3(x) col 3.

W_Matrix =	**$d_1(x)$**	**$d_2(x)$**	**$d_3(x)$**
w0	1.26603335	-0.10584416	-0.16018919
w1	-0.25132905	0.32433516	-0.07300611
w2	0.00983426	-0.65008953	0.64025527

- 34 Misclassified points with 77.33 % training accuracy



-