Exploring SL Approaches for Classifying Iris Plants

Carson Young [1]

Abstract

Two supervised learning approaches on classifying the flowers within the Iris (Fisher, 1988) dataset. I will be comparing and contrasting both k-nearest neighbors and perceptrons (utilizing the typical training rule) in their effectiveness and utility classifying this data.

1. Domain

1.1. History

The history of the Iris data set is both verbose and fraught with issues and errors. Recorded initially by Robert Fisher, a British polymath, it has over time become quite the phenomenon in statistics and machine learning. The data itself is one of the earliest known datasets utilized to test classification methods, and yet it is difficult to prove which (if any) of the datasets are validly and truthfully the original dataset. This is of relatively little consequence for this report specifically, but this does mean that countless famous articles comparing over the ”same” dataset may not have been comparing the same data at all! (Bezdek et al., 1999) This would have some implication for this report, if I had not manually verified that the data from the reports I cite are the same as the data I used.

1.2. Source Code

The source code, direct output (though also, clearly included in this report), and any available documentation can all be found here. No outside influence, packaging, or environments were used to create this project. Just some Python, the dataset, and typical libraries.

[1]Department of Computer Science, University of Oklahoma. Correspondence to: Carson Young <major.add.9@ou.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1.3. Dataset

1.3.1. General Information

The Iris dataset can be formalized as five columns of data: four general characteristics of the flower, and the target ”classification” of the flower.
The four general characteristics, target classification, as well as a photo for clarification, are as follows:

Sepal length/width: the length/width of the sepal area of the flower. The sepal area is the part of the Iris flower many would confoundingly, but not traditionally incorrectly, call the ”petal”.

Petal length/width: the length/width of the petal area of the flower. The petal area is the part of the Iris flower that would be mistaken for leaves if not for the fact that they are typically coloured purple.

Classification: one of three species of Iris flower, Iris-setosa, Iris-versicolor, or Iris-virginica.

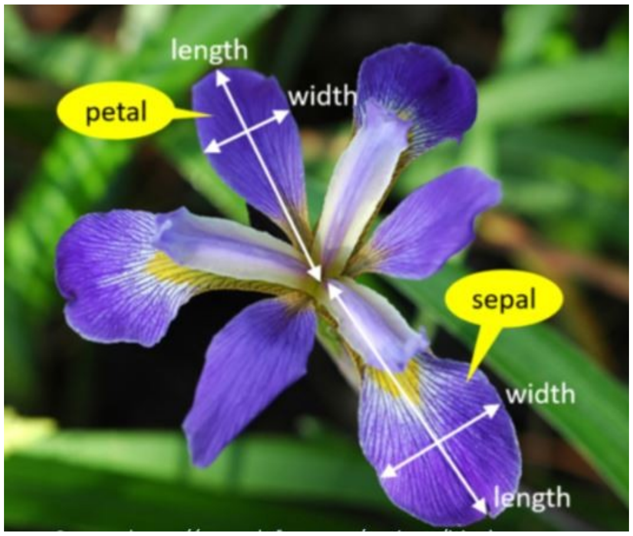

Figure 1. Iris flower with clarifying information about petal vs. sepal areas.

1.3.2. Specific Issues

Though it is the goal of this project to correctly classify Iris flower species based off of the four characteristics above, this will prove incredibly difficult, as only one of the species (Iris-setosa) is actually linearly separable (rather, in many ways, separable at all) from the other two. This is probably the reason that this dataset is still functionally famous in these communities after nearly a century, as if it were well-oriented, well-centered, separable data, then there would be no problem to solve. The other main issue is the breadth of the data. In order to effectively showcase (in my opinion) how these two methods truly operate, I have been processing four dimensions of data. This ends up in a rather startling implication throughout, but I thought it important to note that my struggles with the data could be due to my own ineptitude.

2. Hypotheses

Hypotheses are as follows:

1. The nearest neighbors algorithm will outperform the perceptron due to the perceptron's necessity (by the nature of the thing) for linearity (and linearly separable data).
2. The number of nearest neighbors will reach some number k at which the classification will cease improvements, with the number k being less than the number of samples in the set.

3. Experiments

3.1. General Information

For all hypotheses and all approaches implemented, they all were performed and carried out over the same data set, and the same four-dimensional sample range. As can be seen in Figure 2, the Iris-setosa species is linearly separable across nearly every permutation, but the other two species are not. We can also note from this initial plot, general ranges for each of the four features:

$$\text{Sepal length} \longrightarrow [4.2, 8.0]$$
$$\text{Sepal width} \longrightarrow [2.0, 4.4]$$
$$\text{Petal length} \longrightarrow [1.0, 7.0]$$
$$\text{Petal width} \longrightarrow [0.0, 2.6]$$

With each of these ranges, we can discretize the continuous range while creating a hyperrectangle with each perpendicular feature axis containing points on their respective range, each increasing by 0.2.

I also note that the k-nearest neighbors algorithm is utilizing L^2 (Euclidean) distance for its calculations, and the perceptron rule's/gradient descent's η parameter was held at the value 0.1 for all experiments involving it.

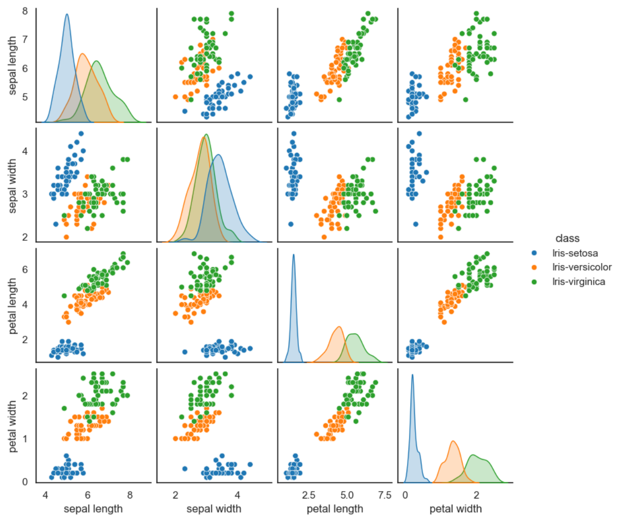

Figure 2. General grid plot showing all permutations of the features, and the coloured classification targets

3.2. Hypothesis 1

3.2.1. Perceptrons

Utilizing a perceptron here was difficult, as it was not a simple two-class problem. Multi-classification with perceptrons involves finding linear separations across *groups* of classifications, then finding the separations via the linear modeling of the perceptron. In this case it means a creating an individual perceptron, one each, for

1. Iris-setosa vs. Iris-versicolor/virginica,
2. Iris-versicolor vs. Iris-setosa/virginica,
3. Iris-virginica vs. Iris-setosa/versicolor,

then comparing the linear coefficients and values to each other. This can be visualized in Figures 3 and 4, in a simplified, two-dimensional example utilizing the same code for the perceptron that I use for this project.

The perceptron's utility not only in this project, but across a variety of situations became clear when it converged to some linear separation of the first species (Iris-setosa) and the other two (Iris-versicolor/virginica) in just 3 iterations. The lack thereof became clear when it would use thousands of iterations for the other two, all while being unable to classify them accurately. This more negatively impacted its

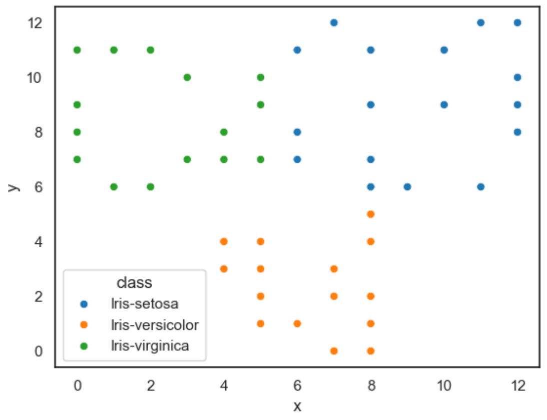

Figure 3. Sparse example data to show general perceptron convergence in multiclass settings

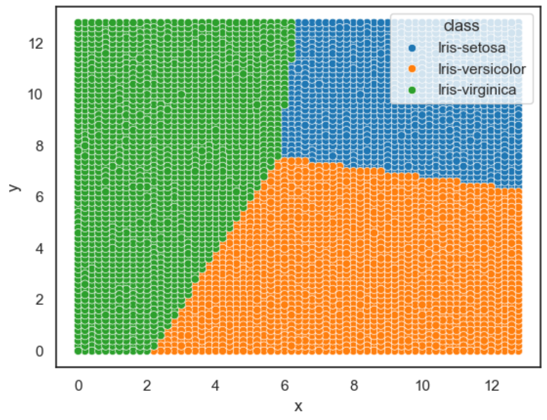

Figure 4. Example multiclass perceptron convergence

ability to register the first species, that of the actually linearly separable data. The perceptron's focus on attempting to find some linear separation of the non-separable data began to negatively impact its ability to separate the initial data. The ineffectiveness can be visualized well in Figure 5, where hardly any Iris-setosa plants are even identifiable (according to the perceptron).

3.2.2. k-Nearest Neighbors

k-nearest neighbors as a classification method for this problem is neither implementation-heavy nor impractical. Actually, the very notion seems to lend itself quite handily to this problem. The visualization shown in Figure 6 using 5-nearest neighbors seems to show clear degrees of classification that appear to match the original data relatively succinctly.

3.2.3. Conclusions

As predicted, the k-nearest neighbors algorithm significantly outperformed the perceptron, as it does not require linear separation to form layers of classification.
Perhaps more intriguingly, though, the implementation of k-nearest neighbors is just intuitive enough to recognize an interesting correlation between its graph and the perceptron convergence graph. It appears that over the four-dimensional space that there are certain planes that appear (at least) partially removable from this problem, as their influence is too topologically simple to help classify anything at all. These planes that I am referring to are the the petal width/sepal width plane, and the sepal length/sepal width plane. This seems to suggest that there is little correlation between these planes of the hyperrectangle, and classification of the Iris species.

3.3. Hypothesis 2

Similarly to above, we can simply visualize the increasing values of k and see if there is any significant change in the visualization. As can be seen in the aforementioned Figure 6 as well as in Figures 7 to 9, there is no significant increase in the algorithm's classification accuracy at all. It remains quite stagnant throughout massive increases of its neighbor count.

3.3.1. Conclusions

My hypothesis was correct, though, I think for the wrong reasons. I hypothesized there was some theoretical upper bound on k that when reached, would not show significant improvement thereafter. This may be true, though, with hindsight, this observation feels rather obvious. It is much more interesting to deliberate whether or not lower values of k would perhaps increase the classification performance, due to the nonlinear combinations of the data (specifically of the latter two species, versicolor and virginica).

4. Literature Review

4.1. Enhanced Classification Models for Iris Dataset

This approach sees the authors compare a range of approaches to that of their implementation, random forests of grafted decision trees. Using LDA (Linear Discriminant Analysis) as their decision boundary, and pre- and post-pruning to reduce variance error were able to increase the AUC over 16% higher (Devasena et al., 2011) than their implementation of k-nearest neighbors, showcasing the quality of their classification model.

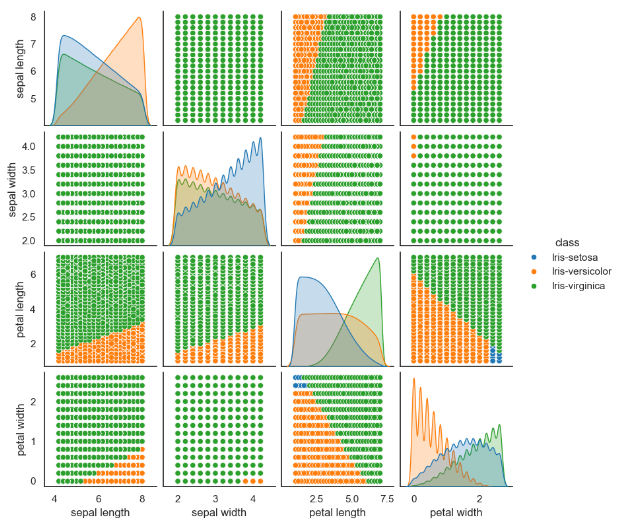

Figure 5. Multiclass perceptron convergence.

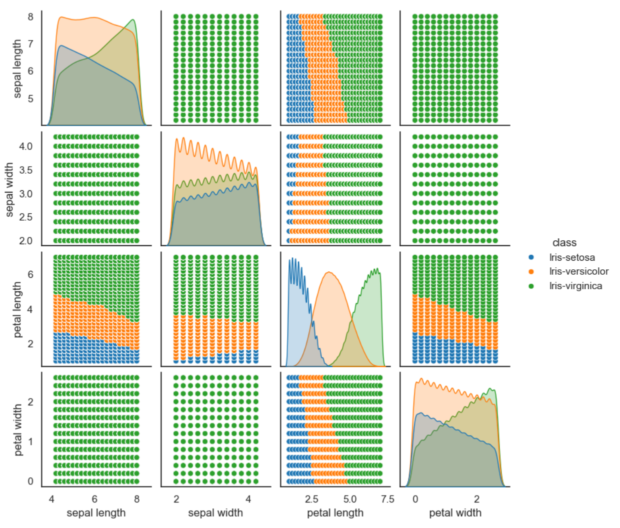

Figure 6. 5-nearest neighbors prediction model for the data hyperrectangle

4.2. An Approach for Iris Plant Classification Using Neural Network

This approach instead sees the authors implement a feed-forward ANN, putting to use a type of neural network used quite extensively in pattern recognition, due to its lack of looping from outputs back to inputs, allowing the ANN to simply find matches of inputs to outputs (Devasena et al., 2011). This implementation saw, from 500 to 50000 epochs, a classification accuracy range of 83.33% to 96.66%.

5. Conclusion

Throughout this project, I have shown the effectiveness (or lack thereof) of perceptrons and the k-nearest neighbors algorithm for classifying flower species from the Iris dataset. Along the way I learned and found much more than this, guiding my interests from linear separability to topological algebras. This problem is, at its core, a question about nature: something so natural, so intuitive; and yet all the while so elaborate and complex that despite generations of scientists and mathematicians, we still struggle to explain some of the simplest aspects of it.

6. Future Work

These implementations, both, are incredibly inefficient. I want to research matrix representations and linear optimizations to aid in the general working cost of my code, as right now, it could use it. If it is possible to drastically cut down the run time (as I am sure it is), then there will be much to do in the way of fiddling with parameters, experimenting with different values, and finding more empirical ways of showing the quality of the classification models.

7. Contributions

The contributions in terms of code and visualization are all mine, but I would be remiss not to thank Dr. Diochnos for his contributions to my thought-processes and support throughout.

References

Bezdek, J. C., Keller, J. M., Krishnapuram, R., Kuncheva, L. I., and Pal, N. R. Will the real iris data please stand up? *7(3), 368-369*, 1999.

Devasena, C. L., Sumathi, T., Gomathi, V., and Hemalatha, M. Effectiveness evaluation of rule based classifiers for the classification of iris data set. *Bonfring International Journal of Man Machine Interface*, 1:5, 2011.

Fisher, R. A. Iris. UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C56C76.

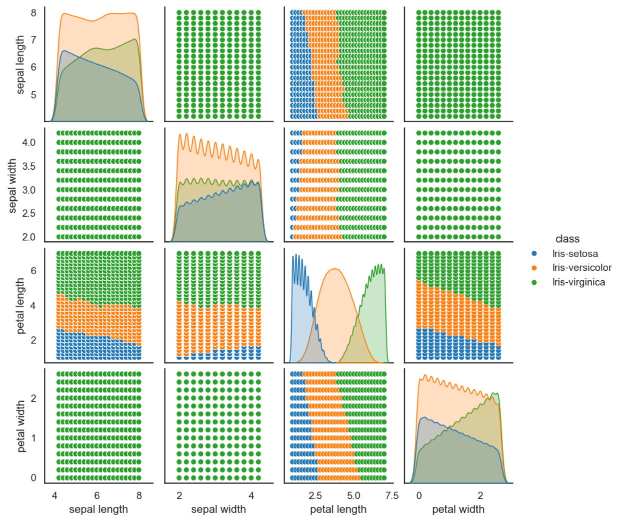

Figure 7. 13-nearest neighbors prediction model for the data hyperrectangle

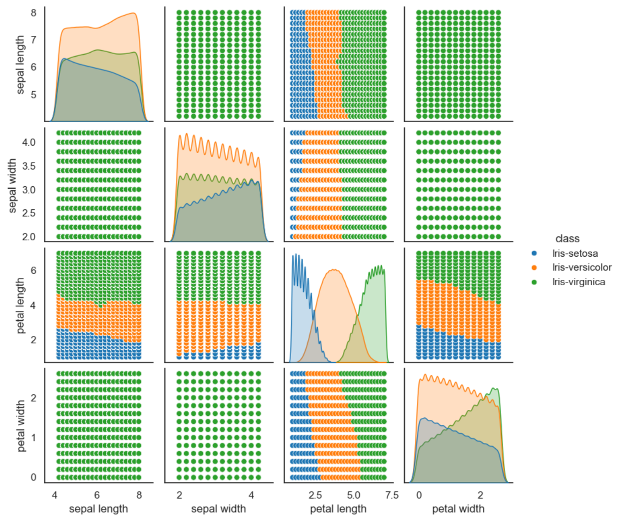

Figure 8. 23-nearest neighbors prediction model for the data hyperrectangle

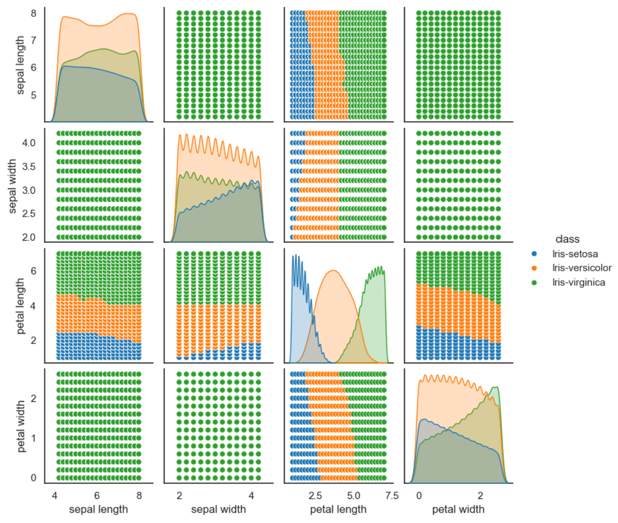

Figure 9. 47-nearest neighbors prediction model for the data hyperrectangle